



EÖTVÖS LORÁND TUDOMÁNYEGYETEM

INFORMATIKAI KAR

Média- és Oktatásinformatika Tanszék

Általános órarend tervező alkalmazás

Dr. Horváth Győző

Egyetemi adjunktus

Tóth Péter

**Programtervező Informatikus
BSc**

Budapest, 2019

Tartalomjegyzék

Bevezetés	1
Felhasználói dokumentáció.....	2
Exodus leírás.....	2
Célközönség.....	2
Rendszerkövetelmény.....	2
Telepítési útmutató	3
Bejelentkezés nélküli funkciók.....	4
Bejelentkezés	4
Exodus	5
Általános felhasználók.....	5
Exodus	6
Kijelentkezés.....	6
Órarend kívánságok	6
Órarend	10
Admin felhasználók	13
Épület adatok	13
Tárgyak	16
Felhasználók	20
Nevek.....	23
Órarend összes tárgya	25
Órarend generálása	28
Fejlesztői dokumentáció	32
Megoldási terv	32
Fő célkitűzések	32
Adatbázis tervezete.....	32
Backend tervezete	40

Frontend tervezete.....	53
Komponensek kommunikációja	62
Megvalósítás	63
Szerver	63
Adatbázis megvalósítás.....	63
Backend	64
Frontend	65
Összegzés.....	66
Tesztelés.....	67
Rendszer teszt	67
Modul teszt	71
Terheléses teszt.....	72
Hatékonyság tesztelése	73
A program jövője	73
Irodalomjegyzék	75

Bevezetés

A programnak az Exodus nevet választottam. A dokumentum további részében, így fogok hivatkozni rá. Az Exodus egy órarend tervező alkalmazás, amelyet általános iskolák használhatnak. Nemcsak a diákok szempontjait veszi figyelembe, hanem a tanárok igényeit is és ennek megfelelően alakítja ki az órarendet.

Minden tanár meg tudja nézni a saját, aktuális órarendjét, illetve megtekintheti az előző félévékét is. Ezenkívül megnézheti, bármely kolléga és osztály órarendjét is. A tanárok megadhatják, hogy melyik osztálynak milyen órát, milyen időpontban szeretnék tartani.

Az alkalmazás adminjai új felhasználót vehetnek fel, jelszót cserélhetnek és admin jogot adhatnak, illetve generálják az órarendet. Ők rögzítik a tantárgyakat, szükség esetén a kigenerált órarendet szerkeszthetik is. Az adminok felvihetnek termeket és a meglévő adatokat módosíthatják. Az alkalmazásban megjelenő nevek is változtathatóak. Meglévő tárgyak mellé újakat is rögzíthetnek, az osztályok számára osztálytermeket jelölhetnek ki. Az adminok mindent felül tudnak bírálni a felületen.

Miért pont egy órarend tervező programot választottam? Érdekes kihívásnak tartottam, hogy lehet algoritmizálni egy órarendet. Kis gondolkozás után rájöttem, hogy ez nem egy egyszerű feladat. Ez csak tovább növelte bennem a lelkesedést, hogy ezt az alkalmazást fejlesszem. Rá kellett jönnöm, hogy nem létezik tökéletes órarend, csak olyan, ami közel áll hozzá. A diákok szemszögéből lehet egy órarend megfelelő ütemezésű, de mindenképpen figyelembe kell venni a tanári oldalt is. Egy tanár több osztályt tanít, ezért előfordulhat, hogy a tanári leterheltség nem egyenletes, mert sok lyukas órája keletkezik (például reggel 8 órától tart egy órát és utána már csak dél után vannak óráit). Ezért próbáltam csinálni egy olyan algoritmust, ami a diákok mellett a tanárok számára is megfelelő eloszlású órarendet hoz létre elkerülve az egyenlőtlen leterheltséget.

Másik oldalról a régi emlékeimből is építkeztem, mély nyomot hagyott bennem az általános iskolai órarend összeállítása. A tanárban egy táblán post-it-ekkel jelölték az órákat és rakosgatták, hogy kinek mikor jó. Ezzel a programmal az ilyen módszerrel órarendet készítő iskolákon is szeretnék segíteni.

Összegezve, azért választottam ezt a témát, mert egy hasznos és kihívásokkal teli programot szerettem volna létrehozni.

Felhasználói dokumentáció

Exodus leírás

Az Exodus egy órarend tervező program, amely figyelembe veszi a tanárok igényeit. Két lépcsős felhasználó kezeléssel működik. Vannak sima felhasználók, akik megtekinthetik a felhasználók és osztályok órarendjeit és felvihetik, hogy mikor szeretnének órát tartani. Az adminok a fentiekén túlmenően képesek órarendet generálni, tárgyakat és termeket rögzíteni, szerkeszteni és a kigenerált órarendet is módosíthatják, valamint a felhasználókat is kezelik.

Célközönség

A program olyan általános iskoláknak készült, amelyeknek nehézséget okoz az órarend megtervezése vagy nagyon sok időbe kerülne elkészíteni. Ha szeretnének egy modern és igényes megoldást találni ennek a problémának a kezelésére, akkor ez a program a tökéletes választás számukra.

Rendszerkövetelmény

A rendszerhez szükséges egy ubuntu 18.04 vagy bármilyen linuxos szerver, amire a következők feltelepíthetők:

- PHP7.2melyhez fel kell telepíteni: mysqlnd, opcache, pdo, xml, calendar, ctype, curl, dom, exif, fileinfo, ftp, gd, gettext, iconv, json, mbstring, mysqli, pdo_mysql, phar, posix, readline, recode, shmop, simplexml, sockets, sysvmsg, sysvsem, sysvshm, tokenizer, wddx, xdebug, xmlreader, xmlwriter, xsl.
- MYSQL adatbázis melynek valamilyen 5.x-es verzióval.
- PYTHON 2.7.15rc1 szükséges a mysql.connector csomag telepítéssel.

A telepítéshez szükséges a MySQL Workbench és valamilyen MySQL adatbázis kezelő program. Javasolt program a HeidiSQL. A szerveren megtalálhatónak kell lennie valamilyen szövegszerkesztőnek pl nano vagy bármilyen, olyan programnak, amivel a szerverre tudunk SFTP-ni és az ott található fájlokat képes szerkeszteni.

Telepítési útmutató

Először győződjünk meg arról, hogy a rendszerkövetelmények teljesülnek a szerveren. Majd hozzunk létre egy új MySQL adatbázist. A mellékeltben megtalálható `mysql/dbstructura.mwb` MySQL Workbench fájlból kigeneráljuk az adatbázist létrehozó SQL-t. Fontos, hogy a MySQL Workbench állítsd át az adatbázis elérést a te adatbázisodra. Ha rossz az SQL, amit kigenerál a program, akkor győződjünk meg róla, hogy jó MySQL verzió van beállítva a programban. Ez után állítsd be, hogy az ID mezők automatikusan növekedjenek. Majd töltsd fel a `T_NAMES` táblát a mellékletben található `mysql/tnamesinser.spl` file segítségével. Ha sikerrel jártunk, akkor `T_USER` táblába létrehozunk egy felhasználót, aminek az `ADMIN` mezőjébe 1-est írunk, a `'SUBJECT_MAIN'` és `'SUBJECT_MINOR'` mezőket nyugodtan hagyjuk `NULL`-on. Ez lesz az első adminja a rendszernek.

Ha ezzel kész vagyunk, akkor a szerverünkre tegyük fel egy olyan elérésű mappába, amihez mindenkinek van hozzáférése és jogosultsága futtatni a mellékletben található `pythonCode/hi.py` fájlt. Ezt után nyissuk meg ezt a fájlt és a `selectSubject`, `selectTeacher`, `selectRoom`, `selectNames`, `selectUser`, `deletActualYears`, `insertTimetableToTTimetable`, `selectTeacherUpper`, `selectSubjectUpper` függvényekben írjuk át az adatbázis kapcsolódáshoz való adatokat az általunk korábban létrehozott adatbázis adataira.

Ha ezzel megvagyunk, akkor kezdjük bele a PHP-s rész telepítésének. A szerveren olyan helyre tegyük a mellékletben szereplő `phpCode` mappát, ahonnan elérjük a `hi.py` kódot. Ha `apach2` használunk, akkor a `var/www/html` mappából hozzunk létre egy symlinket a `phpCode/web` mappára. Egyéb esetben abból a mappából mutassunk rá, ami be van állítva a server configjába kívülről elérhető mappának (`public` mappának). Ha szeretnénk, akkor állítsuk be a domaint a `web` mappára. Fontos, hogy a `web` mappára mutassunk és ne a `phpCode` mappára, mert így csökkentjük az oldal feltörésének az esélyét. Győződjünk meg róla, hogy elérjük az oldalt a megfelelő webcímen. Ha nem érjük el, akkor a jogosultságokkal lesz probléma, ilyenkor a megfelelő mappát `chmod`-dal tudjuk átállítani a jogosultságokat. Ha nem ez a baj, akkor győződjünk meg róla, hogy minden PHP-s követelményt feltelepítettünk. Ezután állítsuk be a `phpCode/config/db.php` fájlba az adatbázis elérést. Majd a `phpCode/views/timetable/generate.php`-ban állítsuk át a `'python /home/xszerion/exodus/pythonCode/hi.py'` szöveget a `'python <hi.py elérési útvonala>'`-

ra. Ha ezzel megvagyunk, jelentkezünk be a korábban létrehozott admin felhasználóval és neki is állhatunk az adatok felvitelének.

Bejelentkezés nélküli funkciók

Bejelentkezés

A bejelentkezést a program kezdő oldaláról tudjuk elérni a bejelentkezés gombbal, amely az 1. ábrán pirossal keretezett helyen található.



Exodus

Üdvözlök az Exodus alkalmazásban. Ez az alkalmazás órarend tervezésre készült. Ha Te az Apáczai Nevelési Központba dolgozol, akkor kérlek, menj tovább és jelentkezz bel Ha még nem rendelkezel felhasználó névvel, vagy elfelejtetted a jelszavadat, akkor a toth.peter.com@gmail.com e-mail címen kaphatsz segítséget. Ha Te nem ebben az iskolában dolgozol, akkor kérlek, hagyd el ezt a weboldalt!

1. ábra. - Bejelentkezés helyére irányítás

Ha átmegyünk a bejelentkező felületre és belekattintunk a felhasználónév mezőbe és kikattintunk belőle, akkor a felület figyelmeztet, hogy a 'Felhasználónév nem lehet üres'(lásd 2. ábra). Ha ugyanígy teszünk a jelszó mezőben, akkor a következő hibaüzenetet kapjuk: 'Jelszó nem lehet üres.'(lásd 2. ábra) ezt úgy tudjuk eltüntetni, ha kitöltjük a megfelelő mezőt.

Felhasználónév:	<input type="text"/>	Felhasználónév nem lehet üres.
Jelszó:	<input type="password"/>	Jelszó nem lehet üres.
<input type="button" value="Bejelentkezés"/>		

2. ábra - Üresen hagyott mező

Amennyiben kitöltöttük a felhasználó nevet és jelszót, de azt a hibaüzenetet kaptuk, hogy 'Rossz jelszó vagy felhasználónév' (lásd 3. ábra), akkor rossz felhasználónevet vagy jelszót írtuk be. Esetleg mind a kettő hibás. Ilyenkor próbáljunk meg újra bejelentkezni helyes felhasználó névvel és jelszóval.

Felhasználónév:

Jelszó:

Rossz jelszó vagy felhasználónév.

Bejelentkezés

3. ábra - Hibás bejelentkezés

Előfordulhat, hogy megjegyzi a felhasználó egy bejelentkezett oldal elérését és azon próbálja meg elérni az alkalmazást. Abban az esetben, ha nincs bejelentkezve, akkor az alkalmazás a bejelentkező felületre továbbítja. Ilyenkor be kell jelentkezni az alkalmazásba, ha ez sikerül, akkor az alkalmazás az eredetileg beírt url-re fogja irányítani a felhasználót és nem az alap kezdőoldalra.

Ha sikerült helyesen beírni a felhasználó nevet és jelszót, akkor beléptünk az alkalmazásba és megjelenik az üdvözlő oldal (lásd 4. ábra). (Kivéve a fent említett esete, amikor az eredetileg beírt url-re irányítja a felhasználót és nem az üdvözlő oldalra.)



Üdvözlöm az Exodusban kedves Bármí Áron

4. ábra - Sikeres bejelentkezés

Exodus

Ha az 1. ábrán az Exodus feliratra kattintunk, akkor újra tölti az index oldalt. Kiváló arra az esetre, ha a kezdő oldal csak részlegesen tudott betölteni.

Általános felhasználók

Az általános funkciók a fejlécen érhetőek el a felhasználók számára sikeres bejelentkezést követően. Ezek az 5. ábrán láthatóak. A fejlécen kékkel jelöli a felület, hogy melyik funkcióban vagyunk.



5. ábra - Általános funkciók

Exodus

Az első funkció az Exodus feliratra kattintva érhető el, amely újra tölti az üdvözlő oldalt (4. ábra).

Kijelentkezés

A második funkció a Kijelentkezés. Ha a kijelentkezés gombra kattintunk, akkor az applikációból kijelentkezünk és a program főindex (1. ábra) oldalára leszünk irányítva.







Órarend kívánságok

A következő funkció az Órarend kívánságok feliratra kattintva érhető el. Ekkor a Kívánságok fognak megjeleni a képernyőn (lásd 6. ábra)

Kívánságok

Új kívánság létrehozása

1-20 megjelenítése a(z) 131 elemből.

#	ID	Félév	Tárgy	Nap	Óra	Osztály	Felhasználó	
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
1	2	2018/19-1	Alsós-reál	Hétfő	2	1.a	user1	  
2	3	2018/19-1	Alsós-reál	Hétfő	3	1.a	user1	  

6. ábra - Kívánságok

A Kívánságok oldalon táblázatos formában megjelennek az eddig felvitt kívánságaink. Az első oszlop címében a # jel látható, itt a táblázat sorainak száma jelenik meg. A következő oszlopban az ID felirat található, amely a kívánság SQL táblájának az azonosítóját jelöli. A Félév felirat alatt az látható, hogy melyik félévre vonatkozik a kívánságunk. A Tárgy felirat alatt az, hogy milyen tantárgyat szeretnénk oktatni. A Nap és Óra oszlopban található, hogy melyik nap, melyik órájában kívánjuk az aktuális órát tartani. Az Osztály alatt szerepel, hogy melyik osztálynak szeretnénk órát tartani. A Felhasználó oszlopban láthatjuk, hogy melyik felhasználó tartja az órát. (Megjegyzés: az adminjoggal nem rendelkező felhasználók csak a saját kívánságaikat látják.). A következő oszlopban három ikon található, melyekhez különböző funkciók tartoznak. Ezeket majd a későbbiekben részletezem.

Most a táblázatban való rendezésről szeretnék beszélni. Ha bármelyik oszlop fejlécében lévő névre kattintunk, akkor az abban az oszlopban szereplő adatok alapján abc sorrendbe rendezi a táblázatot. Ezt egy kis ikon jelezi (lásd 7. ábra). Ha még egyszer rákattintunk a névre, akkor fordított abc sorrendbe rendez, melynek az ikonja a 8. ábrán látható. Ha még egyszer rákattintunk a névre, akkor megint abc sorrendbe rendezi és így tovább. Ha valamelyik oszlop szerint abc sorrendbe van rendezve a táblázat és mi rákattintunk egy másik oszlop címsorában szereplő névre, akkor aszerint az oszlop szerint lesz abc sorrendbe a táblázat.



7. ábra - *Abc rendezés*



8. ábra - *Fordított abc rendezés*

Ha keresni szeretnénk a táblázatban azt is könnyen megtehetjük. A táblázat címsora és az első sora között látunk egy sort, ami tele van üres négyzetekkel, lásd a 6. ábrán. A keresés a következő képpen működik. Valamelyik üres négyzetbe elkezdünk gépelni egy szót (Megjegyzés: Az oszlop neve legyen pl. x oszlop amelyikben gépeltünk.) és kilépünk a cellából, akkor a táblázatban csak azok a sorok fognak megjelenni, melyek tartalmazzák az x oszlopba begéplet szövegrészletet. Például ha az Osztály oszlopba beírom hogy '1.' akkor az összes, olyan sor meg fog jelenni ahol 1. osztállyal kapcsolatos kívánságaim vannak. Akkár több oszlopba is tudunk egyszerre keresni. Ilyenkor csak azokat a találatokat fogja felhozni a táblázat amelyek minden keresési feltételnek eleget tesznek. Ha olyan oszlopban keresek, aminek az értéke csak szám lehet és mi szöveggel próbálunk keresni, akkor a felület felhívja erre a figyelmünket (lásd 9. ábra). Ekkor nem fogja a szűrést végrehajtani. A szűréseket csak akkor hajtja végre, ha az oszlop értékeinek a típusával megegyező értékeket keresünk. Ha a keresés végén nincs találat, akkor olyan kifejezésre szűrtünk, aminek nincsen megfeleltethető sor. Azon oszlopokban, ahol csak számokat lehet megadni ott csak teljes egyezőségre szűrünk. A szűrt táblázaton is működik a rendezés funkció

Óra

fgds

Óra csak egész szám lehet.

9. ábra - *Szöveg hibaüzenet*

Mint korábban említettem a táblázat szélén van 3 kis ikon (lásd. 6. ábra). Most ezt a három ikon funkciót fogom bemutatni. Ha a szemre kattintunk, akkor megnézhetjük az aktuális kívánságot, melyet a 10. ábra prezentál.

Szerkesztés	Törlés
ID	20
Félév	2018/19-1
Tárgy	Angol
Nap	Hétfő
Óra	1
Osztály	1.a
Felhasználó	user2

10. ábra - Kívánság nézet

Itt két további funkciót látunk. Az első a szerkesztés. Ez teljes mértékben megegyezik azzal, ha a 6. ábrán a ceruzát ábrázoló ikonra kattintunk. Ilyenkor bejön a szerkesztő felület az aktuális adatokkal feltöltve melyet a 11. ábra mutat be.

Kívánság szerkesztése:

Félév	2018/19-1
Tárgy	Angol
Nap	Hétfő
Óra	1
Osztály	1.a
Felhasználó	user2
<input type="button" value="Mentés"/>	

11. ábra - Kívánság szerkesztése

A 11. ábrán az adatok címei megfelelnek a 6. ábrán látható táblázat címsorával, a kitöltésüknél az ott leírtak szerint járjunk el. Fontos, hogy az adatokat úgy szerkesszük, hogy a mi kívánságunknak megfeleljen. Figyelembe kell venni, hogy vannak megszorítások az Óra mezőre. Az Óra mező nem lehet üres, ezt a 12. ábrán látható módon jelzi a felület. Ezen kívül csak egész számot fogad el értéknek, amit a 13. ábrán látunk. Figyeljünk rá, hogy 8 óránál ne írjunk többet, mert akkor nagyon értelmetlen órarendet kaphatunk.

Óra

Óra nem lehet üres.

12. ábra - Üres óra mező

Óra

Óra csak egész szám lehet.

13. ábra - Nem szám figyelmeztetés az óránál

Ha elmentjük a szerkesztést a 11. ábrán látható mentés gombbal, akkor a nézet felületre továbbít az oldal.

A következő funkció a törlés. Ha a 11. ábrán a törlésre vagy a 6. ábrán a kuka ikonra kattintunk, akkor az adott elemet töröljük. Törlés előtt felugrik egy figyelmeztető üzenet (lásd 14. ábra) és csak ennek elfogadása után tudjuk törölni az adatokat. Ha a mégse gombra kattintunk, akkor megszakítjuk a törlés folyamatát. Ha törölünk, akkor a 6. ábrára irányít a felület.

A(z) localhost közlendője
Biztosan törölni szeretnéd?

OK Mégse

14. ábra - Törlés figyelmeztetés

Ha a 6. ábrán az Új kívánság létrehozására kattintunk, akkor ugyanaz történik mintha az 5. ábrán az Órarend kívánság hozzáadása feliratra kattintanánk. Bejön egy a szerkesztő felülettel megegyező felület. De vannak különbségek. Itt a legördülő listák értékei az alapértelmezett értékekkel lesznek feltöltve (lásd 15. ábra).

Kívánság létrehozása

Félév

Tárgy

Nap

Óra

Osztály

Felhasználó

Mentés

15. ábra - Új kívánság

A kívánság létrehozásánál ugyanazokat a hibaüzeneteket kaphatjuk, mint a kívánság szerkesztésénél, ezért ezeket nem fogom részletezni. Ha sikerült új kívánságot létrehozni, akkor a 6. ábrára ugrik az applikáció.

Lehetőség van más felhasználó számára is kívánságot létrehozni, ha a kollégáddal megbeszélitek, hogy együtt szeretnétek egy osztályt tanítani, akkor elég csak az egyik félnek a kívánságokat rögzíteni. Nagy segítség lehet azoknak a tanároknak, akik annyira nem értenek a számítógéphez, ezáltal más is felveheti nekik a kívánságokat. Így nem terheljük az adminokat plusz munkával.

Órarend

A felület másik fő funkciója, hogy a felhasználó megnézheti az órarendjét. Ezt úgy tehetjük meg, ha az 5. ábrán az Órarend fülre kattintunk, akkor alapértelmezetten a 16. ábra fog betöltődni.

Órarend

Félév:

Osztály:

Tanár:

Irkérés

Alapértelmezetten a bejelentkezett felhasználó 2018/19-1 félévi órai jelennek meg a táblázatban

#	Hétfő	Kedd	Szerda	Csütörtök	Péntek
0					
1	Informatika Felső info terem user20 8.b 8.a	Informatika Felső info terem user20 8.b 8.a	Informatika Felső info terem user20 5.b 5.a	Informatika Felső info terem user20 5.b 5.a	Informatika Felső info terem user20 6.b 6.a
2	Informatika Felső info terem user20 6.b 6.a	Környezetismeret Meg Terem user20 5.a	Környezetismeret Prorok terem user20 5.b		
3	Környezetismeret Farkas terem user20 6.a	Környezetismeret Prorok terem user20 5.b			
4	Környezetismeret Farkas terem user20 6.a	Informatika Felső info terem user20 7.b 7.a			
5	Környezetismeret Farkas terem user20 6.a	Informatika Felső info terem user20 7.b 7.a	Környezetismeret Tan terem user20 6.b		
6	Környezetismeret Meg Terem user20 5.a	Környezetismeret Tan terem user20 6.b	Környezetismeret Tan terem user20 6.b		
7					
8					

16. ábra - Alapértelmezett órarend oldal

Ennek a tetején látható 3 legördülő lista és egy lekérés gomb. Alatta pedig egy táblázat, amelyben a bejelentkezett felhasználó 2018/19-1. félévi órái láthatóak.

A fenti lekérés úgy működik, hogy a 3 legördülő listából ki választjuk a megfelelő adatokat, majd a lekérés gombra kattintunk. Ha a felhasználó nem választ ki osztályt és tanárt a legördülő listákból, akkor egy üres táblázatot fog visszakapni, lásd 17. ábrán.

#	Hétfő	Kedd	Szerda	Csütörtök	Péntek
0					
1					
2					
3					
4					
5					
6					
7					
8					

17. ábra – Rossz órarend hívás

Ha osztályt és évet választ ki, akkor (lásd 18. ábra) az adott osztálynak a kiválasztott időszak órarendjét fogja megjeleníteni a lekérés gomb.

#	Hétfő	Kedd	Szerda	Csütörtök	Péntek
0					
1	Német Német terem user18 2.b Angol Angol terem user17 2.b	Matematika Kiss Eperke terem user10 2.b	Matematika Kiss Eperke terem user10 2.b	Matematika Kiss Eperke terem user10 2.b	Informatika Kocka terem user19 2.b
2	Matematika Kiss Eperke terem user10 2.b	Matematika Kiss Eperke terem user10 2.b	Matematika Kiss Eperke terem user10 2.b	Matematika Kiss Eperke terem user10 2.b	Környezetismeret Kiss Eperke terem user10 2.b
3	Vizuális kultúra Kiss Eperke terem user10 2.b	Vizuális kultúra Kiss Eperke terem user10 2.b	Testnevelés és sport Kiss Eperke terem user10 2.b	Testnevelés és sport Kiss Eperke terem user10 2.b	Testnevelés és sport Kiss Eperke terem user10 2.b
4	Tánc és mozgás Kiss Eperke terem user10 2.b	Tánc és mozgás Kiss Eperke terem user10 2.b	Magyar nyelv és irodalom Kiss Eperke terem user13 2.b	Magyar nyelv és irodalom Kiss Eperke terem user13 2.b	Erkölcstan/hittan Kiss Eperke terem user13 2.b
5	Ének-zene Kiss Eperke terem user13 2.b	Ének-zene Kiss Eperke terem user13 2.b	Magyar nyelv és irodalom Kiss Eperke terem user13 2.b	Magyar nyelv és irodalom Kiss Eperke terem user13 2.b	Technika, életvitel és gyakorlat Kiss Eperke terem user13 2.b
6					
7					
8					

18. ábra – 2. b. 2018/19-1 órarendje

Ha félévet és felhasználót választunk ki, akkor annak a felhasználónak az adott félévi óráit látjuk (lásd 19. ábra). Ez hasznos lehet helyettesítés esetében, mert leellenőrizhető, hogy szabad-e az a kollégád, akit meg szeretnél kérni, hogy helyetted tartson órát.

Ezenkívül azt is vissza tudjuk nézni ennek a lekérdezésnek a segítségével, hogy korábbi félévekben ki melyik osztályt tanította.

Félév:

Osztály:

Tanár:

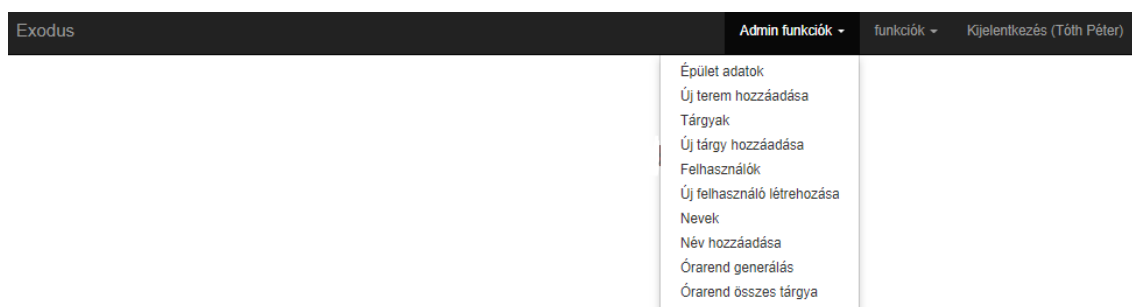
#	Hétfő	Kedd	Szerda	Csütörtök	Péntek
0					
1	Matematika Kiss Eperke terem user10 2.b	Matematika Kiss Eperke terem user10 2.b	Matematika Kiss Eperke terem user10 2.b	Matematika Kiss Eperke terem user10 2.b	Matematika Kiss Eperke terem user10 2.b
2	Matematika Kiss Eperke terem user10 2.b	Matematika Kiss Eperke terem user10 2.b	Környezetismeret Kiss Eperke terem user10 2.b	Vizuális kultúra Kiss Eperke terem user10 2.b	Vizuális kultúra Kiss Eperke terem user10 2.b
3	Testnevelés és sport Kiss Eperke terem user10 2.b	Testnevelés és sport Kiss Eperke terem user10 2.b	Testnevelés és sport Kiss Eperke terem user10 2.b	Tánc és mozgás Kiss Eperke terem user10 2.b	Tánc és mozgás Kiss Eperke terem user10 2.b
4					

19. ábra - User10 2018/19-1 órarendje

Ha mind három mezőt kitöltjük, akkor a program csak az adott osztály aktuális félévi órarendjét fogja visszaadni, mivel nem láttam értelmét, hogy egy osztálynak egy adott tanárára tudjunk szűrni. Ezt a funkciót a böngészővel is elérhetjük a ctrl+f billentyűk lenyomásával.

Admin felhasználók

Az admin felhasználók rendelkeznek az általános felhasználók összes funkciójával, annyi különbséggel, hogy ha a Kívánság táblázatot hozzák be, akkor nem csak a saját, hanem az összes felhasználó kívánságát láthatják. Ezen kívül még a 20. ábrán az Admin funkciók alatt található funkciókkal rendelkeznek.



20. ábra - Admin funkciók

Épület adatok

Az Épület adatok funkcióhoz hozzátartozik az Új terem felvitele funkció. Ha az Épület adatokra kattintunk, akkor a 21. ábra fog bejönni.

Terem adatok

Új terem felvitele

1-20 megjelenítése a(z) 28 elemből.

#	Terem azonosítója	Terem neve	Férőhelyek száma	Specialitás	Osztályterem	
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
1	1	Kis manó	35	-	1.a	 
2	2	Pacal	38	-	2.a	 
3	3	Cseperedők	40	-	3.a	 
4	4	Kiss	35	-	4.a	 

21. ábra- Terem adatok

Ebben a táblázatban a szűrés és a rendezés ugyanúgy működik, mint a Kívánságok táblázatában. A kuka ikon és a nézet törlésnél ugyanúgy megjelenik a figyelmeztető üzenet és annak elfogadásával tudunk törölni. Láthatjuk, hogy a táblázatban más oszlop nevek vannak. Ez a felület az iskola termeinek listázására, szerkesztésére és új terem felvitelére szolgál. A táblázat első oszlopában ismét # szimbólum látható, ez hasonlóan, mint a Kívánság táblázatnál, a sorok számát jelöli. A következő oszlopban a Terem azonosítója szerepel, ami minden sorban különböző. Az órarend generáló program ezt az

azonosítót használja a terem azonosítására. A következő oszlopban a Terem neve található. Az itt szereplő név fog megjelenni a kigenerált órarendben. A következő oszlopban a Férőhelyek számánál azt látjuk, hogy mekkora a terem kapacitása. A következő oszlop a Specialitás. Ez a terem egyedi tulajdonságát jelöli. Pl. így tudjuk megkülönböztetni az infó termet egy sima tanteremtől vagy nyelvi teremtől. Fontos, hogy ha egy órának speciális igényt állítunk be, akkor legyen olyan terem, ami megfelel annak. A következő oszlopban az látható, hogy kinek az osztályterme. Itt ügyelni kell arra, hogy minden alsós osztálynak osszunk osztálytermet, mivel az algoritmus igényli. A felsős osztályoknál nem kell osztálytermet kijelölni, de célszerű, mivel akinek nincs osztályterme annak a speciális tulajdonság nélküli termék közül oszt be egyet a program. Ügyeljünk arra, hogy a termekben elférjenek a csoportok, hisz ha 20 főt akarunk berakni angol órán a 15 angol nyelvi laborba, akkor azt nem tudja beosztani az algoritmus.

Ha rákattintunk a szemre, akkor az adott sor adatai megjelennek. (lásd 22. ábra)

Kis manó

<div>Szerkesztés</div> <div>Törés</div>	
Terem azonosítója	1
Terem neve	Kis manó
Férőhelyek száma	35
A terep specialitása	-
Osztályterme	1.a

22. ábra - Osztály terem nézet

Ha a szerkesztésre vagy a 21. ábrán a ceruzára megyünk, akkor megjelenik a szerkesztő felület a kiválasztott terem adataival (23. ábra).

Terem: Kis manó

Terem neve	<input type="text" value="Kis manó"/>
Férőhelyek száma	<input type="text" value="35"/>
A terep specialitása	<input type="text" value="-"/>
Osztályterme	<input type="text" value="1.a"/>
<input type="button" value="Mentés"/>	

23. ábra - Terem szerkesztése

Itt vannak megszorítások. Ha a terem neve üres, akkor a következő hibaüzenetet kapjuk (lásd 24 ábra): 'Terem neve nem lehet üres.'. Hasonló hibaüzenetet kapunk, ha a férőhelyek számát nem töltjük ki (lásd 25. ábra): 'Férőhelyek száma nem lehet üres'. Ezt úgy tudjuk elkerülni, ha kitöltjük a mezőt. De vigyázzunk, mert ha szöveggel töltjük ki számok helyett, akkor ezt a hibaüzenetet kapjuk (lásd 26. ábra): 'Férőhelyek száma csak egész szám lehet'. Ezért a férőhelyek számát számmal adjuk meg. Az osztályoknál, ha nem szeretnénk, hogy valakinek az osztályterme legyen, akkor válasszuk a vonalat, egyéb esetben, ha osztályteremnek szeretnénk, akkor válasszuk a megfelelő osztályt. Ha van valamilyen speciális tulajdonsága a teremnek, akkor a specialitásnál válasszuk ki a megfelelőt, ha nincs, akkor válasszuk a vonalat. Ha minden sor ki van töltve és nincs hibaüzenet, akkor el tudjuk menteni a változtatásokat és a program átirányít a nézet földre, ahol le tudjuk ellenőrizni a változásokat.

Terem neve	<input type="text"/>
Terem neve nem lehet üres.	

24. ábra - Üres teremnév

Férőhelyek száma	<input type="text"/>
Férőhelyek száma nem lehet üres.	

25. ábra - Üres Férőhely szám

Férőhelyek száma	<input type="text" value="asd"/>
Férőhelyek száma csak egész szám lehet.	

26. ábra - Nem szám a férőhelyek száma

Ha a címsorban az új terem hozzáadása gombra kattintunk vagy a 21. ábrán az új terem felvitele gombra, akkor be fog jönni a 27. ábra, ami majdnem ugyanaz, mint a szerkesztés csak a mentéssel a végén nem egy adott termet szerkesztünk, hanem újat viszünk fel. Így ennek a működését tovább nem részletezem. Ha nem szeretnénk létrehozni új termet, akkor a címsorból válaszuk ki azt a funkciót, amivel folytatni szeretnénk a munkát és egyszerűen elhagyhatjuk ezt az aloldalt.

Terem felvitele

Terem neve

Ferőhelyek száma

A terep specialitása

Osztályterme

Mentés

27. ábra - Új terem



Tárgyak

A Tárgyak funkcióhoz hozzá tartozik az új tárgy hozzáadása funkció. Ha a Tárgyakra kattintunk a 20 ábrán, akkor a 28. ábra fog bejönni.

Tárgy adatok

Új tárgy felvitele

1-20 megjelenítése a(z) 203 elemből.

#	ID	Félév	Név	Osztály	Osztály létszáma	A tárgy nehézségi szintje	Speciális igény	Óraszám	Társóra	
1	1	2018/19-1	Matematika	1.a	30	Nehéz	-	8	0	 
2	2	2018/19-1	Magyar nyelv és irodalom	1.a	30	Nehéz	Dupla	5	0	 
3	3	2018/19-1	Tánc és mozgás	1.a	30	Könnyű	-	3	0	 
4	4	2018/19-1	Technika, életvitel és gyakorlat	1.a	30	Könnyű	-	2	0	 
5	5	2018/19-1	Testnevelés és sport	1.a	30	Könnyű	-	1	0	 

28. ábra - Tárgy adatok

Ezen a felületen a rendezés és a szűrés funkció ugyanúgy működik, mint a Kívánságok táblázatnál. Itt az Osztály létszáma és az Óraszámnál van szám ellenőrzés a szűrésben (lásd. 29. ábra) Ugyanúgy kezelendő, mint a Kívánságoknál az Óra oszlop.

Osztály létszáma	A tárgy nehézségi szintje	Speciális igény	Óraszám
<input type="text" value="asd"/> <small>Osztály létszáma csak egész szám lehet.</small>	<input type="text"/>	<input type="text"/>	<input type="text" value="asd"/> <small>Óraszám csak egész szám lehet.</small>

29. ábra - Szám ellenőrzés

A táblázat fejlécében az osztályok tervezett óráinak az adatai és jellemzői szerepelnek. A # jelű oszlop a táblázat sorszámát tartalmazza. Az ID alatt az adott órák azonosítója szerepel, ami az adatbázisban az órát azonosítja. A Félév azt mutatja, hogy melyik félévre vonatkozik az adott tárgy. A Név oszlop a tantárgy nevét tartalmazza. Az Osztály alatt látható, hogy melyik osztály tárgya. Az Osztály létszám oszlop tartalmazza, hogy az adott osztályból hány fő vesz részt az adott órán. A tárgy nehézségi szintje alatt látható, hogy milyen nehéz egy tárgy. Minél nehezebb egy tárgy annál korábbi időpontra lesz beosztva az óra. Speciális igénynél látjuk, hogy dupla legyen-e az óra vagy valamilyen speciális típusú teremben kell megtartani. Az Óraszám a heti óraszámot takarja. Társóra oszlopba az az óra szerepel, amivel egy időben kell megtartanod az órát. A nézet felület ugyanúgy működik, mint azt már korábban bemutatam, így arra nem térek ki. (A nézet felületet lásd 30. ábra).

szerkesztés
törlés

ID	1
Félév	2018/19-1
Név	Matematika
Osztály	1.a
Osztály létszáma	30
Tárgy nehézségi szintje	Nehéz
Speciális igény	-
Óraszám	8
Társóra	0

30. ábra - Tárgy nézet

A tárgy szerkesztése a 31. ábrán látható. Ezen a felületen fordul elő először, hogy van olyan mező ahová nem kötelező adatot megadni. Ez a Társóra cellája. Ha nullát írunk be, az ugyanazt jelenti mintha nem írtunk volna be semmit. A társóránál nagyon fontos, hogy létező óra ID-ját állítsunk be, amelyet a táblázatból ki tudunk keresni. Ha nem létező ID-t állítunk be, akkor az órarend generálásánál nem az elvárt eredményt fogjuk kapni. A társóra lehet másik osztály órája is, illetve a saját osztályunkat is csoportokra bonthatjuk. Ilyenkor nagyon ügyeljük arra, hogy az Osztály létszámnál a bontott csoportok létszámát adjuk meg, különben a termék kiosztása nem lesz optimális. (A szükségesnél nagyobb termékekbe lesznek beosztva a csoportok.)

Egy egyszerű példával szemléltetem a használatát:

Legyen az 1-es az azonosítója az 1.a angolórájának, ahova 15 járnak. Az 1.a másik fele a 2 azonosítójú német órára jár. Ilyenkor a 2 azonosítójú németórának beállítjuk a 1 ID-t társórának. Ha még azt is szeretnénk, hogy az 1.b osztály angolórája együtt legyen a 1.a angolórájával, akkor beállítjuk az 1.b osztály angol órájánál az 1 azonosítót társóra azonosítónak. Ha még azt is szeretnénk, hogy az 1.b németórája együtt legyen az 1.a németórájával, akkor beállítjuk a 2 azonosítót társórának a 1.b német órájánál. Ekkor egy időpontban lesz a 4 óra és egy tanár fogja tartani a közös angolórát és egy másik tanár a közös németórát. Fontos, hogy ilyenkor az az óra, aminek nincsen beállítva testvér órája, annak az óraszám szerinti mennyiségű közös óra lesz létrehozva.

Tárgy szerkesztése

Félév	2018/19-1
Név	Matematika
Osztály	1.a
Osztály létszáma	30
Tárgy nehézségi szintje	Nehéz
Óraszám	8
Speciális igény	-
Társóra	
Mentés	

31. ábra - Tárgy szerkesztése

A Tárgy szerkesztése felületen az Óraszám és a Társóra cellába szöveget nem lehet megadni. (Szám a mező elvárt értéke.) Szöveg beírásakor hibaüzenetet kapunk. (lásd. 32. ábra.) Az Óraszám mező nem lehet üres, kötelező elem, ha nem töltjük ki, akkor a 33. ábrán látható hibaüzenet jelenik meg. Ezek a hibaüzenetek megszűnnek, ha számot írunk a cellákba. Felhívom a figyelmet, hogy amíg van hibaüzenet, addig nem lehet elmenteni a változtatásokat. (A korábbi formoknál is él ez a szabály.)

The screenshot shows a form titled 'Tárgy szerkesztése'. It has three main sections: 'Óraszám' with a text input containing 'asd' and an error message 'Óraszám csak egész szám lehet.' below it; 'Speciális igény' with a dropdown menu showing '-'; and 'Társóra' with a text input containing 'asd' and an error message 'Társóra csak egész szám lehet.' below it.

32. ábra - Nem szám hibaüzenet a tárgyaknál

The screenshot shows the 'Óraszám' section of the form. The text input is empty, and an error message 'Óraszám nem lehet üres.' is displayed below it.

33. ábra - Üres óraszám hibaüzenet

Ha a 20. ábrán látható Új tárgy hozzáadása feliratra, vagy a 28. ábrán lévő Új tárgy felvétele gombra kattintunk, akkor a 34. ábra fog megjelenni a képernyőn, ami ugyanúgy viselkedik, mint a tárgy szerkesztése form csak itt nem egy meglévőt szerkesztünk, hanem egy új tárgyat viszünk fel. Az értékei az alapból beállított értékek lesznek. Az értékekre ugyanazok a megszorítások vonatkoznak, mint a szerkesztésnél.

Tárgy létrehozás

The screenshot shows the 'Tárgy létrehozás' form. It contains several input fields: 'Félév' (dropdown with '2018/19-1'), 'Név' (dropdown with 'Alsós-humán'), 'Osztály' (dropdown with '1.a'), 'Osztály létszáma' (text input), 'Tárgy nehézségi szintje' (dropdown with 'Könnyű'), 'Óraszám' (text input), 'Speciális igény' (dropdown with '-'), and 'Társóra' (text input). At the bottom left is a green 'Mentés' button.

34. ábra - Tárgy létrehozása











Felhasználók

A 20. ábrán a Felhasználók és az Új felhasználó létrehozás gombok egybe tartoznak. Ha Felhasználókra kattintunk, akkor a 35. ábra fog megjelenni.

Felhasználók

Felhasználó létrehozása

1-20 megjelenítése a(z) 32 elemből.

#	Felhasználónév	Jelszó	Felhasználó teljes neve	Fő tárgya	Másodlagos tárgya	Admin	
	<input type="text"/>	<input type="password"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
1	user1	123456	Kiss Pista	Alsós-reál	Alsós-humán	0	 
2	user10	123456	Bármí Áron	Alsós-reál	Alsós-humán	0	 
3	user11	123456	Bor Virág	Alsós-reál	Alsós-humán	0	 
4	user12	123456	Bornemissza Ivó	Alsós-reál	Alsós-humán	0	 
5	user13	123456	Boviz Huba	Alsós-humán	Alsós-reál	0	 

35. ábra – Felhasználók

Itt az összes felhasználót látjuk. A törlés, szűrés, rendezés funkciók ugyanúgy működnek, mint az előző táblázatoknál. Itt csak az Admin sorra van szám szűrés beállítva. Az első oszlop, hasonlóan a korábbi táblázatokhoz # jelű és a táblázat sorsszámát tartalmazza. A Felhasználó név az elsődleges azonosító, ezért ennek egyedinek kell lenni. Ezzel tud a felhasználó belépni. A Jelszó a felhasználó jelszavát mutatja. A Felhasználó teljes neve oszlopban a felhasználó nevét láthatjuk. Ha bejelentkezik, ez jelenik meg az üdvözlő oldalon és az oldal címsorában is. A Fő tárgya a tanár elsődleges tantárgya a melléktantárgya, pedig a tanár Másodlagos tárgya. Ezek az oszlopok akár üresek is lehetnek, ha csak egy admint szeretnénk létrehozni. Ha a felhasználó admin, akkor az Admin oszlopban egy 0-tól különböző szám szerepel. Célszerű az egyes számot választani. A nézet behozása ugyanúgy működik, mint korábban a megtekintendő sorban a szemre kattintva (lásd 36. ábra).

Kiss Pista

Szerkesztés Törlés

Felhasználónév	user1
Jelszó	123456
Felhasználó teljes neve	Kiss Pista
Fő tárgya	Alsós-reál
Másodlagos tárgya	Alsós-humán
Admin	0

36. ábra - Felhasználó nézet

A Felhasználó szerkesztése felületet a 37. ábra szemlélteti. Itt az Admin kivételével egyik mező sem lehet üres, (lásd. 38. ábra) illetve az admin mezőben csak számok szerepelhetnek, ahogy ezt a 39. ábra mutatja. Ezeknek a hibaüzeneteknek a megszüntetése a helyes adatok beírásával lehetséges. Ha egy felhasználónak szeretnénk új jelszót beállítani, akkor itt tudjuk ezt megtenni. Ha az Admin mezőbe üresen hagyjuk, akkor nem lesz admin jogosultságú a felhasználó.

Felhasználó szerkesztése: Kiss Pista

Felhasználónév

Jelszó

Felhasználó teljes neve

Fő tárgya

Másodlagos tárgya

Admin

Mentés

37. ábra - Felhasználó szerkesztése

Felhasználó szerkesztése: Kiss Pista

Felhasználónév

Felhasználónév nem lehet üres.

Jelszó

Jelszó nem lehet üres.

Felhasználó teljes neve

Felhasználó teljes neve nem lehet üres.

38. ábra - Üres felhasználó adatok figyelmeztetés

Admin

Admin csak egész szám lehet.

39. ábra - Nem szám hibaüzenet az adminál

Ha új felhasználót szeretnénk létrehozni, akkor a 20. ábrán szereplő Új felhasználó létrehozás felíratra vagy a 35. ábrán látható Felhasználó létrehozása gombra kattintunk. Ekkor a korábbiakhoz hasonlóan a Szerkesztéssel megegyező felületet kapunk az alap értékekkel feltöltve (lásd 40. ábra). Itt szeretnék megemlíteni egy plusz ellenőrzést, ami a Szerkesztésnél is létezik, de ennél a felületnél kiemelt jelentőséggel bír. Ha olyan felhasználónevet választunk ki, ami már foglalt, akkor a 41. ábrán látható hibaüzenetet kapjuk miután a mentés gombra kattintottunk. Ezt a hibát megoldhatjuk úgy, hogy megnézzük a 35. ábrán látható táblázatot és keresünk egy olyan felhasználónevet, ami még nem létezik, vagy próbálgatjuk, hátha találunk egy nem foglalt felhasználónevet. Érdemes konvenciót alkalmazni, hogy a keresztnéve + a vezetéknéve első betűje, ha az foglalt, akkor a következő betűt hozzáadni vagy elkezdni számozni a felhasználókat. Így könnyebben elkerülhetjük a felhasználónevek ütközését.

Felhasználó létrehozása



Felhasználónév

Jelszó

Felhasználó teljes neve

Fő tárgya

Másodlagos tárgya

Admin

Mentés

40. ábra - Felhasználó létrehozása



Felhasználó név

user1

Felhasználó név "user1" már használatban van.

41. ábra - Foglalt felhasználónév

Nevek






















A Nevek és a 20. ábrán látható Név hozzáadása funkció egy funkciónak tekintendő mivel a név hozzáadás a Nevek funkció alatt is megtalálható. Az oldal gyorsabb használata miatt szerepeltetjük ezt a funkciót a menüsoron is. Ha a 20. ábrán a Nevekre kattintunk, akkor a 42. ábra jön be.

Nevek

Új név hozzáadása

Ezen az oldalon tudod megnézni, illetve módosítani a felületen megjelenő különböző neveket.

1-20 megjelenítése a(z) 104 elemből.

#	Azonosító	A megjelenő név	A név típusa	
	<input type="text"/>	<input type="text"/>	<input type="text"/>	
1	0	-	EMPTY	  
2	1	Hétfő	DAY	  
3	2	Kedd	DAY	  
4	3	Szerda	DAY	  
5	4	Csütörtök	DAY	  
6	5	Péntek	DAY	  
7	6	Matematika	SUBJECT	  

42. ábra - Nevek

A Nevek táblában azokat a név azonosító párosokat láthatjuk, amik a program felületén fognak megjelenni. Helyüket, hogy hol jelenjenek meg azt a Név típusa szabja meg. A táblázatban a szűrés, keresés és törlés a korábbihoz hasonló módon történik. A táblázat első oszlopa a szokásos # szimbólum, így itt a sorok számát láthatjuk. Mellet az Azonosító oszlopba szerelő értékek a tábla fő kulcsa. Az órarend generátor függvény ezeket használja, ezért vigyázni kell a szerkesztésükkel. A napoknak fontos, hogy az 1, 2, 3, 4, 5 helyen legyenek. Ezen kívül még arra is figyeljünk, hogy a 0 azonosító a vonal legyen, a 93 azonosító pedig a dupla óra. Arra is kiemelt figyelmet fordítsunk, hogy a könnyű, közepes, nehéz azonosítók is ebben a sorrendbe szerepeljenek. Az ezen a felületen történő szerkesztést nagy körültekintéssel végezzük, különben nagy problémákat fogunk okozni. Ez a felület azért van, mert előfordulhat, hogy mégis bele kell nyúlni a nevekbe. A megjelenő név oszlopban az azonosítóhoz tartozó nevek találhatók. A név típusa oszlopban a következő értékek szerepelhetnek: EMPTY, DAY, SUBJECT, CLASS, LEVEL, SPECIAL, YEARS. Az EMPTY azt jelöli, ahova üres értéket szeretnénk beszúrni. Ez fixen a 0 azonosító. A DAY a napokat, SUBJECT a megjelenő tárgyakat jelöli. Ha új tárgyat akarunk felvinni, akkor SUBJECT típust kell választani. A CLASS az osztályt azonosítja, ha új osztályt szeretnénk rögzíteni, akkor ilyen típust kell alkalmazni. A LEVEL a nehézségi szintet jelöli. Ez akkor kell, ha pl. új nehézségi szintet

szeretnénk bevezetni. A SPECIAL egy teremnek a speciális tulajdonságát, a YEARS egy félévet jelöl. Ezt akkor kell szerkeszteni, ha új félévet szeretnénk felvinni.

A nézet itt is a szokás módon működik. A szerkesztésnél csak arra kell figyelni, hogy A megjelenő név mező nem lehet üres, mert akkor a 43. ábrán látható hibaüzenet jelenik meg. Ennek megoldása a név kitöltése.

Szerkesztendő név: Kedd

A megjelenő név

A megjelenő név nem lehet üres.

A név típusa

mentés

43. ábra - Név szerkesztési hiba

Ha a 20. ábrán a Név hozzáadásra vagy a 42. ábrán az Új név hozzáadása gombra kattintunk, akkor szinte ugyanazt a felületet kapjuk, mint a szerkesztésnél csak itt nem egy meglevő nevet szerkesztünk, hanem egy új nevet adunk hozzá az eddigiekhez. (lásd 44. ábra.)

Új név felvitele

A megjelenő név

A név típusa

mentés

44. ábra - Új név felvitele











Órarend összes tárgya

Ha a 20. ábrán az Órarend funkcióra kattintunk a 45. ábrára fogunk ugrani, amin a kigenerált órarend adatai vannak.

Órarend

Óra hozzáadása

1-20 megjelenítése a(z) 462 elemből.

#	Félév	Tárgy azonosítója	Tárgy neve	Osztály	Nap	Óra	Tanár	Szoba	
1	2018/19-1	92	Magyar nyelv és irodalom	4.b	Hétfő	1	user1	László Brigitta te	 
2	2018/19-1	92	Magyar nyelv és irodalom	4.b	Hétfő	2	user1	László Brigitta te	 
3	2018/19-1	92	Magyar nyelv és irodalom	4.b	Kedd	1	user1	László Brigitta te	 
4	2018/19-1	92	Magyar nyelv és irodalom	4.b	Szerda	1	user1	László Brigitta te	 
5	2018/19-1	92	Magyar nyelv és irodalom	4.b	Csütörtök	2	user1	László Brigitta te	 

45. ábra - Generált órarend

Ebben a táblázatban is a korábbiakhoz hasonlóan működik a szűrés, a törlés és a rendezés. Az első oszlopban a szokásos # szimbólumot láthatjuk, ami ebben az esetben is a táblázat sorainak a számát mutatja. A Félév oszlop az időszakot jelöli, hogy melyik félévre vonatkozó adatokat találjuk az adott sorban. A következő oszlop a Tárgy azonosítója, ezek az értékek azonosítják a program számára a tárgyakat. A tantárgyak elnevezése a Tárgy neve oszlopban szerepelnek. Az Osztály oszlop tartalmazza, hogy melyik osztályé, a Nap oszlop, hogy melyik nap lesz az adott óra. Az Óra oszlopban pedig, hogy hányadik órában lesz. A Tanár oszlopban látható, hogy melyik felhasználó tartja az órát. A Szoba oszlopban található, hogy melyik tanteremben van az aktuális óra. Ha szemre kattintunk, akkor bejön a Tárgy nézete, amit a 46. ábrán látunk.

Szerkesztés	Törlés
Félév	2018/19-1
Tárgy azonosítója	92
Tárgy neve	Magyar nyelv és irodalom
Osztály	4.b
Nap	Hétfő
Óra	1
Tanár	user1
Szoba	László Brigitta te
ID	20703

46. ábra - Tárgy nézet

Egy generált tárgy szerkesztésénél nagyon körül tekintőnek kell lenni, mert ennek a lehetőségét a felület megadja, teljes szabadságot biztosít, hogy semmilyen elképzelésnek ne szabjon határt és az adminok szabadon szárnyalhassanak. A szerkesztő felület a 47. ábrán látható.

Szerkesztés

Félév
2018/19-1

Tárgy azonosítója
92

Tárgy neve
Magyar nyelv és irodalom

Osztály
4. b

Nap
Hétfő

Óra
1

Tanár
user1

Szoba
László Brigitta 1e

Mentés

47. ábra – Órarendi óra szerkesztése.

A Tárgy neve, Tárgy azonosítója, az Óra és a Szoba nem lehet üres, erre a felület hibát fog dobni. (lásd. 48. ábra.) Továbbá az Óra és a Tárgy azonosítója csak szám lehet, különben a 49. ábrán látható hibaüzenetet kapjuk. Érdeemes úgy szerkeszteni, hogy mindig csak egy adatot változtatunk meg mivel így mindig könnyen vissza tudjuk ellenőrizni, hogy nem hoztunk létre ütközést.

Tárgy azonosítója
Tárgy azonosítója nem lehet üres.

Tárgy neve
Tárgy neve nem lehet üres.

Osztály
4. b

Nap
Hétfő

Óra
Óra nem lehet üres.

Tanár
user1

Szoba
Szoba nem lehet üres.

48. ábra - Órarendi óra üres mezők hibája

Tárgy azonosítója

Tárgy azonosítója csak egész szám lehet.

Tárgy neve

Osztály

Nap

Óra

Óra csak egész szám lehet.

49. ábra - Tárgy szöveg hiba

A következő lehetőség a 45. ábráról elérhető Óra hozzáadása funkció. Ez majdnem ugyanaz, mint a szerkesztés csak itt nem egy adott órát szerkesztünk, hanem egy újat viszünk fel. A felületet az 50. ábra tartalmazza. Ezt a funkciót akkor használjuk, ha már van egy kész órarendünk, de rájöttünk, hogy az egyik osztálynak egy órája kimaradt és nem akarjuk újra generálni az órarendet. Itt is ugyanazoknak a feltételeknek kell megfelelni, mint a szerkesztésnél.

Óra hozzáadása

Félév

Tárgy azonosítója

Tárgy neve

Osztály

Nap

Óra

Óra nem lehet üres.

Tanár

Szoba

Mentés

50. ábra - Órarendi óra hozzáadása

Órarend generálása

Ha a 20. ábrán látható Órarend generálás gombra kattintunk, akkor az 51. ábra fog megjelenni, amely az órarend generáló felület. Az egész alkalmazásban ez a legegyszerűbb felület, pedig ez a legösszetettebb és legfontosabb.

Órarend generálása

Válassza ki melyik félévet szeretné legenerálni:

2018/19-1

Generálás

51. ábra - Órarend generáló felület.

Kiválasztjuk a listából, hogy melyik félévet szeretnénk kigenerálni és rákattintunk a Generálás gombra. Ha minden rendben van, akkor az 52. ábrán látható üzenetet kapjuk miután újra töltött az oldal.

Órarend generálása

Válassza ki melyik félévet szeretné legenerálni:

2018/19-1

Generálás

Az órarend generálása kész.

52. ábra - Jó órarend készült

Ha például nincs egy adott típusú tanárból elég az iskolában, akkor a következő üzenetet kapjuk (lásd. 53. ábra). Ennek a megoldása, ha viszünk fel megfelelő típusú felhasználót.

Órarend generálása

Válassza ki melyik félévet szeretné legenerálni:

2018/19-1

Generálás

Nincs elég Alsós-humán típusú tanár.

53. ábra Nincs elég tanár

Ha az adott félévre nem hoztunk létre tárgyakat az adatbázisban, akkor 54. ábrát kapjuk. Ilyenkor kell felvinni az osztályoknak tárgyakat.

Órarend generálása

Válassza ki melyik félévet szeretné legenerálni:

2018/19-2

Generálás

Nincsenek tárgyak felvive az adott félévben!

54. ábra - Nincsenek tárgyak

Az 55. ábrán látható hibaüzenet esetében előfordulhat, hogy nem a tanárok számával, hanem a felvitt tárgy adataival van probléma. Ebben az esetben a hibaüzenet végén zárójelben lévő számsor ad segítséget. Az első szám a tárgy azonosítója. Ha minden rendben van a tárggyal, akkor a megfelelő típusú tanárt kell hozzáadni a rendszerhez.

Órarend generálása

Válassza ki melyik félévet szeretné legenerálni:

2018/19-1

Generálás

Nem tudunk ennek a tárgynak tanárt találni, mert túl vannak terelve a tanárok. (69, 46, 20, 97, 87, 1, 17)

55. ábra - Hibás tárgy

Előfordulhat, hogy az 56. ábrán látható hibaüzenetet kapjuk. Ebben az esetben túl van terelve az iskola. Ha lehet, akkor csökkentjük az órák számát, de ha nem tudjuk, akkor próbálkozzunk plusz terem, illetve tanárok hozzáadásával, esetleg csökkentjük az összevont órák számát, ami segítheti megoldani a problémát, de valójában az osztály túlterhelése a fő gond. Mivel alsóban a program egy nap max. 8 órát enged, felsőben pedig 9-et, ezért ha heti 40 óránál is több órájuk van, akkor kiakadhat a program.

Órarend generálása

Válassza ki melyik félévet szeretné legenerálni:

2018/19-1

Generálás

Túl sok órájuk van!

56. ábra - Túl sok órájuk van

Előfordulhat, hogy nem vittünk fel termeket az iskolához. Ilyenkor az 57. ábrán látható hibaüzenetet kapjuk. A megoldásához vigyünk fel termeket.

Órarend generálása

Válassza ki melyik félévet szeretné legenerálni:

2018/19-1

Generálás

Nincsenek termék felvive az iskolához!

57. ábra - Terem nélküli iskola

Elképzeltető, hogy a nevek tábla üres, ilyenkor az 58. ábrán látható hibát kapjuk. Ekkor a mellékelt SQL-el töltjük fel a nevek táblát.

Órarend generálása

Válassza ki melyik félévet szeretné legenerálni:

2018/19-1

Generálás

Nincsenek nevek felvéve az adatbázisból

58. ábra - Nincsenek nevek

Ha nincsenek felhasználók felrögzítve, akkor az 59. ábrán látható hibaüzenet jelenik meg. Vigyünk fel felhasználókat!

Órarend generálása

Válassza ki melyik félévet szeretné legenerálni:

2018/19-1

Generálás

Nincsenek felhasználók felvéve

59. ábra - Nincsenek felhasználók

Előfordulhat, hogy olyan osztálytermet kap az osztály, amibe nem fér el. Ilyenkor a 60. ábrán látható hibaüzenetet kapjuk. Ennek megoldása adjunk akkora termet, amiben elfér az osztály.

Órarend generálása

Válassza ki melyik félévet szeretné legenerálni:

2018/19-1

Generálás

A(z) 1.b létszáma nagyobb mint a(z) Kiss Eperke terem befogadó képessége

60. ábra - Rossz osztályterem

Ha valamelyik tárgynak nem adtunk meg termet, akkor a következő hibaüzenet érkezik. (lásd.61.ábra) Ezt úgy tudjuk kezelni, ha felviszünk megfelelő specialitású termet.

Órarend generálása

Válassza ki melyik félévet szeretné legenerálni:

2018/19-1

Generálás

Az iskolában nem létezik Német típusú terem.

61. ábra - Terem hiány

Ha egy teremben túl sok óra van, akkor 62. ábrán szereplő hibaüzenetet kapjuk. Ilyenkor érdemes erről a teremről néhány osztályterem funkciót eltávolítani.

Órarend generálása

Válassza ki melyik félévet szeretné legenerálni:

2018/19-1

Generálás

A Kocka terem túl van terhelve!

62. ábra - Terem túlterhelés

Ha egy tanár túl sok osztálynak szeretne órát tartani, akkor érkezik a 63. ábrán látható hibaüzenet. Ilyenkor a Kívánságok táblában az adott tanárnak néhány óráját ki kell törölnünk.

Órarend generálása

Válassza ki melyik félévet szeretné legenerálni:

2018/19-1

Generálás

A user1 túl van terhelve!

63. ábra - Túlterhelt felhasználó

Ha nincs elég olyan terem, aminek nincs specialitása, akkor a következő a 64. ábrán látható hibaüzenetet kapjuk. Ilyenkor specialitás nélküli termet kell felvinni

Órarend generálása

Válassza ki melyik félévet szeretné legenerálni:

2018/19-1

Generálás

Szükséges tulajdonság nélküli terem hozzáadása vagy növelni kell a terem befogadó képességét!

64. ábra - Kevés átlagos terem

Fejlesztői dokumentáció

Megoldási terv

Fő célkitűzések

A programtól azt várom el, hogy egy órarendet elő tudjon állítani, ami figyelembe veszi a tanárok kívánságait. Ennek a feladatnak a megoldásához kell egy adatbázis, amiben tároljuk az órarend generálásához és a felület használatához szükséges adatokat. Szükséges egy felület, amely biztosítja a tárgyak, termek, felhasználók, felhasználói kívánságok felvitelét, szerkesztését és törlését. Ezen kívül még meg tudja jeleníteni az órarend generálást és utasítást képes adni új órarend generálására, valamint a kigenerált órarendnek a szerkesztését biztosítja. Szükséges még egy algoritmus, ami kigenerálja a felhasználók kívánságainak figyelembevételével az órarendet. A tervek szerint ebből a három részből fog állni az órarend tervező program.

Adatbázis tervezete

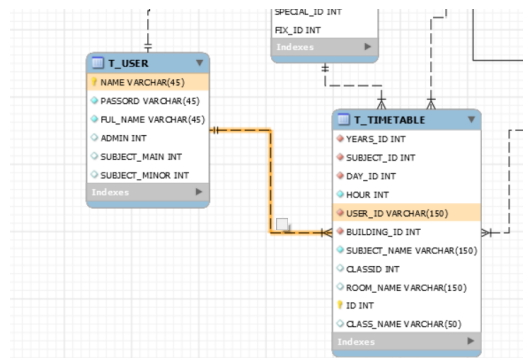
Adatbázis választás

Adatbázis terén felvetődött bennem, hogy Oracle-t kellene használni. Majd kisebb gondolkodás után rájöttem, hogy túlságosan drága a liszensz és az alkalmazásban nem kell bonyolult adatbázis függvényeket használni. Kis nézelődés után a MySQL mellett döntöttem. Ez ingyenes és egyszerű használni. Azt is el kellett döntenem, hogy milyen verziót használjak. Nagyon érdekesnek találtam a 8.0 verziót. Sok újítás van a nyelvben, de ezekre nincs szükségem, ráadásul a MySQL Workbench még nem tudja kezelni a 8-as verziót és kevés forrás található az interneten. Megnéztem és a gépemem lévő MySQL Workbench éppen a MySQL 5. verzióját támogatta, ezért ezt a verziót választottam.

Kapcsolat tervek

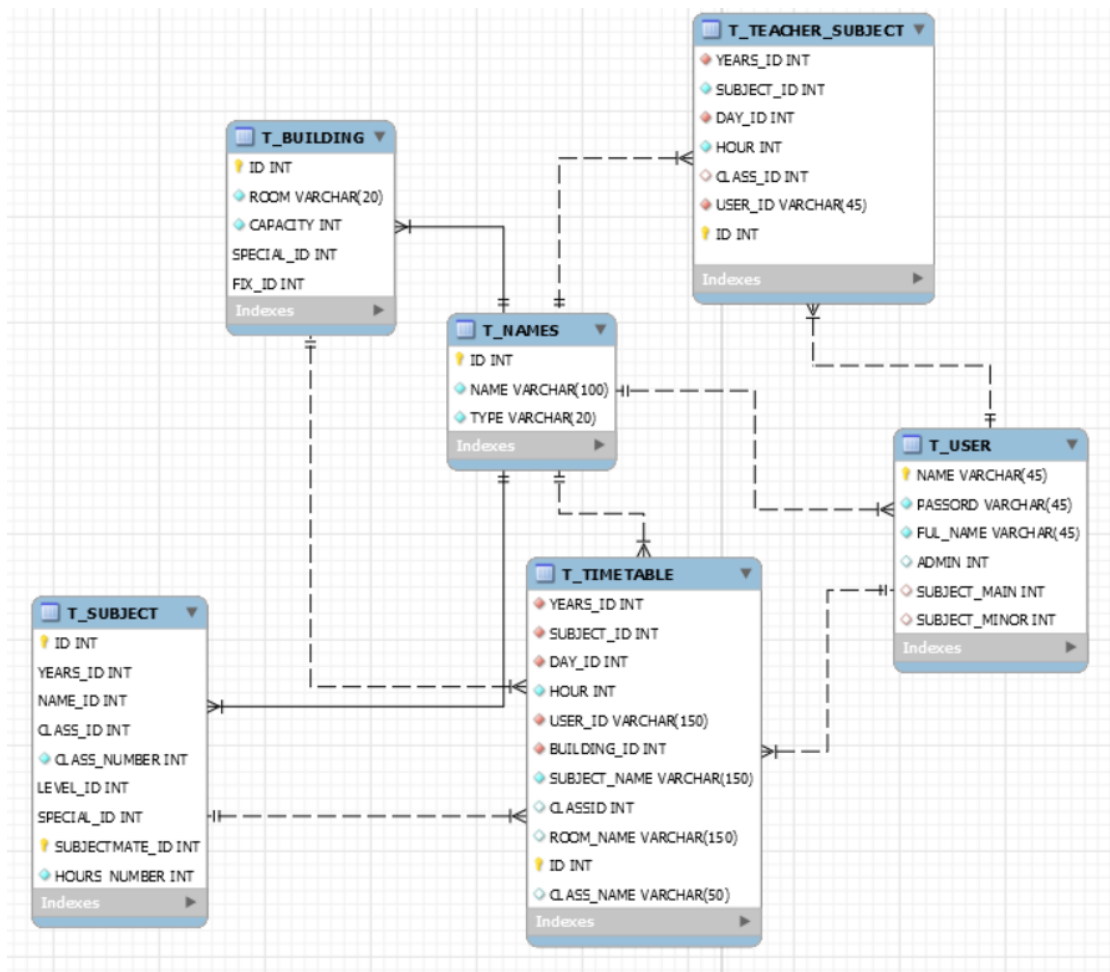
A 66. ábrán látható a tervezett adatbázis kapcsolatok. Ezt az ábrát MySQL WorkBenchben készítettem, melynek forrása megtalálható a mellékelt lemezen a mysql/dbstructura.mwb fájlban. Ha megnyitjuk a fájlt és az egérrel rámegyünk a két táblát összekötő vonalra, akkor a 65. ábrán látható módon megjelenik, hogy mi - mivel van kapcsolatban, illetve a program alsó felében

tábla szinten is kiírja, hogy mi- mivel- milyen módon van kapcsolatban. Mint láthatjuk, a 66. ábrán minden tábla mindennel kapcsolatban van. Nézzük egyesével a kapcsolatokat. Az első tábla legyen a T_USER tábla. Ennek a SUBJECT_MAIN, SUBJECT_MINOR oszlopa függ a T_NAMES ID oszlopától. Mehetünk a következő táblára, ez a



65. ábra - Kapcsolat megjelenítés

T_NAMES tábla, ami az előzőhöz hasonlóan nem függ semmitől. A következő legyen a T_BUILDING tábla. Ez a tábla már függ a T_NAMES táblától. A T_BUILDING és a FIX_ID értékei csak olyanok lehetnek, ami megfeleltethető a T_NAMES ID oszlopának. Ennek a táblának nincsen több függése. A következő tábla a T_SUBJECT. Ennek a táblának a YEARS_ID, NAME_ID, CLASS_ID, LEVEL_ID, SPECIAL_ID függ a T_NAMES ID oszlopától. Csak olyan értékeket vehet fel, ami szerepel az ID oszlopban. Következzen a T_TEACHER_SUBJECT tábla. Ennek a táblának YEARS_ID, CLASS_ID, DAY_ID függ a T_NAMES ID oszlopában található értékektől. Továbbá a USER_ID függ a T_USER NAME oszlopától. Legvégül pedig a T_TIMETABLE tábláról essen szó. A YEARS_ID, DAY_ID függ a T_NAMES ID oszlopától. A BUILDING_ID függ a T_BILDING ID oszlopától, A SUBJECT_ID függ a T_SUBJECT ID -tól. A USER_ID függ a T_USER NAME oszlopától. Vannak táblák ahol még várható lenne kapcsolat, mint a T_TIMETABLE táblánál a CLASSID, de ez nem kell, mivel az osztályt a T_SUBJECT táblán keresztül kapja meg. A többi esetben is más táblán keresztül kapja meg az értékét.



66. ábra - Adatbázis terv

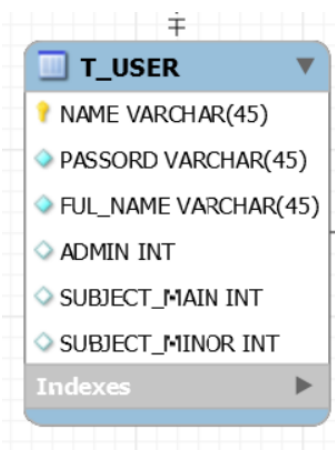
Táblák leírása

Általános megjegyzések

A következő tábláknál az ID nevű mezőkre be kell állítani, hogy automatikusan növekedjenek minden új sor beszúrásánál az adatbázisban. Ezen túl alapértelmezetten egyik mező sem lehet üres. Amelyeknél előfordulhat, azt külön jelzem a táblázat leírásában.

T_USERS

Ebben a táblában a rendszer felhasználóit fogjuk tárolni. A tábla adatai a 67. ábrán láthatjuk. Szükségünk van egy elsődleges kulcsra, ami azonosítja a felhasználót. Ez a NAME oszlop lesz. Ebben az oszlopban 45 hosszú varchar típusú értékek lesznek. Mivel lesz login felület ezért szükség lesz egy mezőre, ahol a jelszót tárolom. Ez a PASSORD oszlop, melyben 45 hosszú varchar típusú értékek lesznek. Azért hogy a felület személyes legyen és tudja üdvözölni a felhasználót, kelleni fog egy olyan oszlop, ahol a felhasználó rendes nevét tároljuk. Ez legyen a FULL_NAME oszlop, mely 45 karakter hosszú varchar típusú értékeket vesz fel. Szükség

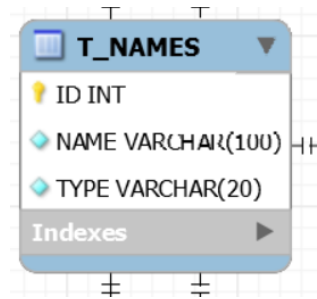


67. ábra -
Felhasználók tábla

lesz, olyan oszlopra ahol letároljuk, hogy milyen típusú órákat oktat. Erre két oszlopot fogunk használni, ahol csak a tárgy azonosítóját tároljuk. A két tárgy elég, mert általában csak ennyit tanítanak együtt a tanárok. Ezek legyenek a SUBJECT_MAIN és a SUBJECT_MINOR. Ezeknek az értékeik legyenek egyszerű int-ek. Kelleni fog egy mező, aminek a segítségével eldöntjük egy felhasználóról, hogy admin-e. Ez legyen az ADMIN mező, amelynek értéke legyen int, mert azt a többi felület egyszerűbben tudja kezelni. Ha 0 vagy üres az értéke, akkor nem admin a felhasználó, ha ki van töltve, akkor admin.

T_NAMES

Szükségünk lesz egy táblára, ahol a felületen tárolt legördülő menüs értékeket tároljuk, hogy ha esetleg többnyelvűsíteni szeretnénk a felületet könnyebben meg tudjuk tenni. Ezen kívül a programok számokat gyorsabban tudnak kezelni, de a felhasználók nem fogják megérteni ezt, ezért szükség lesz a nevek és az azonosítók párosítására. Kell egy oszlop ahol az azonosítókat fogjuk tárolni. Ez legyen az ID oszlop, ami int típusú és ez a tábla elsődleges kulcsa. Szükség lesz egy mezőre ahol az azonosítóhoz társított neveket tároljuk. Ez legyen a NAME és mivel előfordulhatnak hosszú nevek, ezért legyen 100 hosszú varchar típusú. Kell egy típus azonosító oszlop, hogy a felületen le tudjuk szűrni a legördülő menük értékeit. Ez legyen a TYPE, ami 20 hosszú varchar, és a következő értékeket veheti fel: EMPTY, DAY, SUBJECT, CLASS, LEVEL, SPECIAL, YEARS.

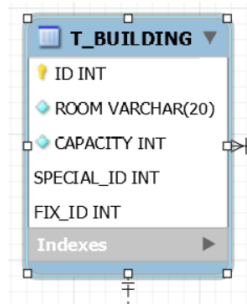


68. ábra - Név tábla

- EMPTY: Ez a 0 ID-hoz tartozzon mivel így tudunk „trükközni” az ürese érték kitöltésével mivel ugye a 0 a hamis érték
- DAY: Ez tárolja a napokat. Fontos, hogy a napok azonosítóik sorrendben az 1, 2, 3, 4, 5 legyen, mivel ezt az algoritmusban könnyen ki tudjuk majd használni a több dimenziós tömbök indexelésénél.
- SUBJECT: Ez az órák neveinek és a tanárok által tanítandó órák jelölésére szolgál.
- CLASS: Az osztályok neveinek a jelölésére szolgál.
- LEVEL: A tárgyak nehézségének a neveinek jelölésére szolgál
- SPECIAL: A termék, illetve alsós tanárok tárgyainak és tárgyak speciális igényeinek a jelölésére szolgál.
- YEARS: Adott félév jelölésére szolgál.

T_BUILDING

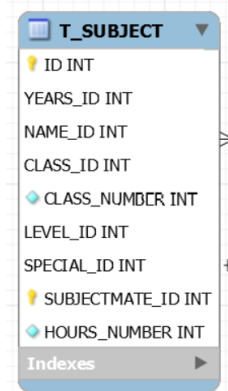
Szükség lesz egy táblára, ahol tárolni fogjuk az épület adatait. Ez lesz a T_BUILDING tábla. Melynek szerkezetét a 69. ábra mutatja be. Kell egy azonosító, amivel az algoritmus beazonosítja. Ez lesz az ID ami int típusú és ez a tábla elsődleges kulcsa. Minden teremnek van neve, a ROOM-ban tároljuk ezeket, ami 20 hosszú varchar típusú értékeket vehet fel. Kell egy mező, ahol eltároljuk, hogy hány főt fog befogadni a terem. Ez legyen a CAPACITY, amely int értékeket vehet fel. Lehet, hogy a termeknek lesznek speciális tulajdonságai pl. tesi terem vagy informatika terem stb. Ezért kellene fog egy mező ahol ezeknek az azonosítóit eltároljuk. Ez a SPECIAL_ID ami int típusú értékeket tartalmaz. Ez a mező akár NULL értéket is felvehet. Előfordulhat, hogy valamelyik teremet osztályteremnek szeretnénk alkalmazni. Ilyenkor el kell tárolnunk az osztály azonosítóját. Ez lesz a FIX_ID ami int típusú értékeket vehet fel és ez is felveheti a NULL értéket.



69. ábra - Épület adatok

T_SUBJECT

Kell egy tábla, ahol eltároljuk az osztályok óráit. Ez a tábla a T_SUBJECT lesz, melyet a 70. ábra prezentál. Szükség lesz egy egyedi azonosítóra, ami azonosítja a tárgyat. Ez lesz az ID, ami int típusú és elsődleges kulcs a táblában. Le kell tudnunk tárolni annak a félévnek az azonosítóját, amelyikben az adott tárgy lesz. Ez a YEARS_ID ami int típusú változókat tartalmaz. Le kell tudnunk tárolni a tárgy nevének az azonosítóját. Ez lesz a NAME_ID ami int típusú értékeket tartalmaz. Ismernünk kell, hogy melyik osztályhoz tartozik az aktuális óra. Ezért az osztályok azonosítóit a CLASS_ID oszlopban tároljuk, amelyek int típusú értékek lehetnek. Tudnunk kell, hogy egy tárgyra hányan járnak, hiszen lehet, hogy osztálybontás van az aktuális tárgyból. Ezt a CLASS_NUMBER oszlopban tesszük, melynek int típusú értékei lehetnek. Ismernünk kell egy tárgyról, hogy milyen nehéz. A nehézségi szint azonosítóit a LEVEL_ID oszlopban tároljuk, aminek az értékei int típusúak. Tudnunk kell, hogy a tárgyaknak van-e valamilyen speciális terem igényük. Ezeknek az azonosítóit a SPECIAL_ID oszlopban tároljuk, melynek az értékei int típusúak. Tudnunk kell egy tárgyról, hogy hány óra van belőle egy héten, hogy ne kelljen az admin felhasználónak



70. ábra - Tárgyak tábla

pl. ötször felvinni egy matekórát. Így könnyebb figyelni arra, hogy egy órát egy tanár vigyen. Ezeket a számokat a HOURS_NUMBER oszlopban tároljuk, melynek értékei int típusúak. Tudnunk kell órákat egy időpontban tartani. Ehhez ismerni szükséges, hogy melyik órával szeretnénk azonos időpontot. A SUBJECTMATE_ID-ban tároljuk a társóra azonosítóit melyek intek és lehetnek üresek is.

T_TEACHER_SUBJECT

Kell egy tábla, ahol a tanárok kívánságait fogjuk letárolni. Ez a tábla a T_TEACHER_SUBJECT mely a 71. ábrán látható. A táblának kell egy egyedi azonosító mivel a tervezet keretrendszer szükségessé teszi az adatok egyszerű megkeresését érdekében. Ez lesz az ID oszlop, amelyben int típusú értékek lesznek és ez a táblázat elsődleges kulcsa. Szükségünk lesz arra az információra, hogy milyen órát szeretne tartani a tanár. A SUBJECT_ID-ban lesz letárolva a tárgy nevének az azonosítója mely int típusú. Tudnunk kell, hogy melyik osztálynak szeretne órát tartani. A CLASS_ID-ban lesz az osztály azonosítója letárolva mely int típusú. Ismernünk kell, hogy melyik felhasználóhoz tartozik kívánság. A felhasználó azonosítóját a USER_ID-ban tároljuk, amelynek a típusa 45 hosszú varchar. Tudnunk kell, hogy melyik nap lesz az óra. A DAY_ID oszlopban fogjuk tárolni az int típusú azonosítóját. Azt is ismernünk kell, hogy hányadik órában szeretné tartani. Ezt HOUR oszlopban tároljuk, mely értékeinek a típusa int.

Column	Type	Primary Key
YEARS_ID	INT	No
SUBJECT_ID	INT	No
DAY_ID	INT	No
HOUR	INT	No
CLASS_ID	INT	No
USER_ID	VARCHAR(45)	No
ID	INT	Yes

71. ábra - Kívánságok tábla

T_TIMETABLE

Kell egy tábla ahol a generált órarendeket fogjuk tárolni. Ez lesz a T_TIMETABLE tábla, aminek a reprezentációját a 72. ábrán látjuk. Szükség lesz egy azonosítóra, ismételten a használt keretrendszer miatt. Ez az ID lesz, ami int típusú. Tudnunk kell, hogy melyik félévhez tartozik, így a félév azonosítóját a YEARS_ID oszlopban tároljuk, mely int típusú. Szükséges tudnunk, hogy melyik tárgy, az azonosítóját a SUBJECT_ID oszlopban tároljuk, mely int típusú. Ismernünk kell, hogy melyik nap lesz. Az int típusú azonosítót a DAY_ID oszlopban tároljuk. Tudnunk kell, hogy melyik tanár tartja az órát. A felhasználó azonosítóját a USER_ID mezőben tartjuk,

Column	Type	Primary Key
YEARS_ID	INT	No
SUBJECT_ID	INT	No
DAY_ID	INT	No
HOUR	INT	No
USER_ID	VARCHAR(150)	No
BUILDING_ID	INT	No
SUBJECT_NAME	VARCHAR(150)	No
CLASSID	INT	No
ROOM_NAME	VARCHAR(150)	No
ID	INT	Yes
CLASS_NAME	VARCHAR(50)	No

72. ábra - Órarend tábla

melynek az értékei 150 hosszú varchar. Szükséges ismernünk, hogy melyik teremben lesz. A terem azonosítóját a BUILDING_ID-ban tároljuk, melynek értékei int típusúak. Tudnunk kell, hogy melyik osztályé az óra. Ezt a CLASSID oszlopban tároljuk és int típusú azonosító. Mivel a felületen az órarendet egy egyedi php függvény fogja összerakni és gyorsítani szeretnénk a felületet ezért nem a lassú PHP fogja összehúzni a név - érték párokat, hanem letároljuk az adatbázisban. Ezért szükséges a tárgy neve, melyet a SUBJECT_NAME-ben fogunk letárolni, ami 150 hosszú varchar típusú értékeket tárol. Ugyanilyen indokkal kell a terem neve, amit a ROOM_NAME-ben fogunk letárolni, melynek elemei 150 hosszú varchar típusúak. Legvégül az osztályok neveit is ilyen módon fogjuk tárolni a CLASS_NAME oszlopban. Az oszlop elemei 50 hosszú varchar típusúak lesznek

Backend tervezete

Programozási nyelv választása

Feltettem a kérdést, hogy melyik nyelvet válasszam. Általam a legismertebb backend nyelv a PHP. Erről hamar eldőlt, hogy nagyon lassú nyelv a többi nyelvhez képest. Majd elgondolkodtam a Javán, de arról az volt a véleményem, hogy túl nagy nyelv és az ottani gondolkozásmód eltér az általam megszokottól. Után felmerült benne, hogy a C, de azt hamar elvettem mivel ott van a C++. Utánanéztem és azt írták több helyen és az ismerősöktől is azt hallottam, hogy az adatbázis kapcsolat nem a leggyorsabb és nagyon bonyolult tud lenni. Ezek után elgondolkodtam, hogy milyen gyors számítású nyelvek léteznek, amik esetleg még szóba jöhetnek. Ekkor jött a felismerés, hogy a PYTHON egy nagyon gyors nyelv. Utánanéztem, de ennek a nyelvnek sem valami nagyon gyors az adatbázis kapcsolata, lassabb is, mint a C++. Viszont vannak előnyei például, hogy nagyon egyszerű az adatbázis kapcsolat. Ezen felül még a program jövőjét illetően van egy hatalmas előnye a PYTHON-nak, nagyon jók a mesterséges intelligencia moduljai. Ez a dokumentáció végén, a program jövője résznél fog értelmet nyerni, hogy ez miért fontos. Emellett még az is sokat számított, hogy megszerettem volna tanulni PYTHON-ban is programozni. Majd felmerült a kérdés, hogy a 2-es vagy a 3-as verziót használjam. A kettesre eset a választás mivel ehhez több mintakód és forrás van a neten.

Elméleti algoritmus felépítése

Mint a bevezetőben is írtam nem egyszerű tökéletes algoritmust írni az órarend tervezésre. Én a következő alapelvekre jutottam. Mivel az alsós és felsős órarend nagyon különböző, ezért kétfelé szedem az algoritmust. Lesz egy alsós és egy felsős órarend tervező algoritmusom, aminek természetesen lesznek közös függvényei. Azt a döntés is meghoztam, hogy ha valamelyik tanár írt kívánságot egy osztállyal kapcsolatban, akkor azt előre veszem a fontossági sorrendben. Így ha valaki szorgalmas és felviszi az igényeit a felületre, akkor sokkal jobb órarendet kap, mint aki nem csinál semmit. Ezeknek az elveknek fog megfelelni a generált órarend.

Alsós órarend

Algoritmus rövid leírása

Lekérjük az adatbázisból az alsósokra vonatkozó adatokat és az általános adatokat. Majd a lekért adatok alapján az osztályokat és a termeket párosítjuk. Aztán ketté bontjuk azokat a tanárokat, akik az általános és azokat, akik a speciális órákat tartják. Majd a tanárok

kéréseit figyelembe véve párosítjuk az osztályokat és a tanárokat. Ezután beállítjuk, hogy az összes társóra az első órára mutasson. Majd összerakjuk őket. Aztán elkezdjük időpontra beosztani a kívánságok alapján a speciális órákat, majd a sima órákat. Ez után készen van az alsós órarendünk.

Megvalósítandó függvények

Adatbázisból olvasandó függvények

selectNames

Bemenete: -

Kimenete:

- names tömb: Melynek egy eleme a T_NAMES táblában látható [ID, NAME, TYPE] hármas.

Feladata: A T_NAMES táblából lekérje az adatokat.

selectSubject

Bemenet:

- years_id: Az aktuális félév azonosítója.

Kimenet:

- subject tömb: Melynek egy elem a T_SUBJECT táblában található [ID, NAME_ID, CLASS_ID, CLASS_NUMBER, LEVEL_ID, SPECIAL_ID, HOURS_NUMBER, SUBJECTMATE_ID] sornak felel meg.

Feladat: Lekéri az órákat a T_SUBJECT táblából nehézségi szint szerint csökkenő sorrendben.

selectTeacher

Bemenet:

- years_id: Az aktuális félév azonosítója.

Kimenet:

- teacher tömb: Ami a T_TEACHER_SUBJECT tábla elemeit tartalmazza. Így egy elem a tömbnek a következő tömb: [SUBJECT_ID, DAY_ID, HOUR, CLASS_ID, USER_ID].

Feladat: Lekéri a paraméterben megkapott félév alsós óráival kapcsolatos kívánságokat a T_TEACHER_SUBJECT táblából.

selectUser

Bemenet: -

Kimenet:

- user tömb: Mely egy eleme a T_USER táblából a [NAME, SUBJECT_MAIN, SUBJECT_MINOR] tömb.

Feladat: Lekéri a T_USER tömbből a felhasználókat.

selectRoom

Bemenet: -

Kimenet:

- room tömb: Melynek egy eleme a T_BUILDING táblából származó [ID, ROOM, CAPACITY, SPECIAL_ID, FIX_ID] tömb.

Feladat: Lekéri a T_BUILDING táblából az épület adatait.

deletActualYears

Bemenet:

- years_id: Az aktuális félév azonosítója.

Kimenet: -

Feladat: Tranzakció kezeléssel ellát törlés, ami a T_TIMETABLE táblából kitörli az aktuális félévre vonatkozó adatokat

insertTimetableToTTimetable

Bemenet:

- timetableRow: A generált órarend egy sora.

Kimenet: -

Feladat: A T_TIMETABLE táblába beszúr egy sor. Fontos, hogy ez is tranzakció kezeléssel ellátott legyen.

Adatok párosítása

createtimetable

Bemenet:

- subject: Az órák adatai.
- room: A termek adatai.

- names: A név és azonosító párok a T_NAMES táblából.

Kimenet:

- timetable tömb: Melynek egy eleme egy tömb ami [az aktuális félév azonosítója, tárgy azonosítója, terem azonosítója, osztály azonosítója].

Feladat: Végig megy a kapott tárgyakon, ha a tárgynak nincs speciális tulajdonsága, akkor az osztályteremet párosítja mellé, különben a tárggyal azonos speciális tulajdonsággal rendelkező termet társít hozzá. Közben ellenőrzi, hogy minden tárgynak tud termet adni és hogy az adott terembe belefér az osztály.

isGoodRooms

Bemenet:

- timetable tömb: Melynek egy eleme egy tömb ami [az aktuális félév azonosítója, tárgy azonosítója, terem azonosítója, osztály azonosítója].
- subject: Az órák adatai.
- room: A termek adatai.

Kimenet: -

Feladat: Végig megy a timetable tömbön és megnézi, hogy semelyik terem és semelyik osztály sem szerepel 40-nél többször az timetable tömbben. Ha igen, akkor hibát dob az adott osztályra vagy teremre.

serachTeacher

Bemenet:

- timetable tömb: Melynek egy eleme egy tömb ami [az aktuális félév azonosítója, tárgy azonosítója, terem azonosítója, osztály azonosítója].
- subject: Az órák adatai.
- teacher: A tanárok kívánságai.
- user: A felhasználók adatai.
- names: A név és azonosító párok.

Kimenet:

- user: Ahol már az alap user adatokon kívül már az osztályok is be vannak osztva.

Feladat: A függvény megnézi, hogy hány fix tanár szeretné tanítani az adott osztályt. Ha kettő, akkor beosztja az adott osztályhoz. Ha kettőnél több, akkor azt a két tanárt

osztjuk be, akik leghamarabb vitték fel az osztály iránti kívánságukat. Ha egy, akkor beosztjuk őt és keresünk neki egy párt. Ha egy sem, akkor keresünk két tanárt. Fontos, hogy a fix tanárok közül csak azokat tudjuk beosztani, akiknek még nincs osztályuk.

UserAddDreamClass

Bemenet:

- subject: Az órák tömbje.
- users: A felhasználók adatai.
- teacher: A kívánságok tömbje.

Kimenet:

- user: A felhasználók adatai órák azonosítóival társítva.

Feladat: A kívánságok alapján hozzárendeljük a felhasználóhoz az órákat.

userAddBadDreamClass

Bemenet:

- subject: Az órák tömbje.
- users: A felhasználók adatai.

Kimenet:

- user: A felhasználók adatai órák azonosítóival társítva.

Feladat: Azokat az órákat, amiket nem szeretne senki sem tartani, azokat osztja be.

settingSubjectmatetoFirstClass

Bemenet:

- subject: Az órák tömbje
- users: A felhasználók adatai

Kimenet:

- subject: Az órák tömbje a társórákkal összevonva
- users: A felhasználók adatai

Feladat: Azokat az órákat, amiket egy időpontban tartunk, összevonjuk és tanárt társítunk hozzá.

Időpont foglalás

Megjegyzés

A következőkben bevezetjük a roomMatrix-t, a teacherMatrix-t és a classMatrix-t, melyeknek az utolsó eleme egy 5x8 mátrix. Ez a mátrix megfeleltethető egy hét beosztásának. Tehát a mátrix[0][1] indexe a hétfő első óráját jelöli. A 0 érték a napot, az 1 értéke az órát jelöli. Mivel a T_NAMES táblában 1-5 azonosítójú elemek a napok, így ha kivonunk belőle egyet, akkor tudjuk indexként használni az értékeit. Az órák értéke 0-7 közé esnek, így azok tökéletesen megfelelnek indexnek. Ha az 5x8 mátrix valamelyik eleme nem nulla, akkor ott, az abban az időpontban szereplő tárgy azonosítója fog szerepelni.

creatTimetable

Bemenet:

- user: A felhasználók adatai és óráik.
- subject: Az alsós tárgyak.
- roomMatrix: Egy mátrix, aminek egy eleme [szoba azonosít, specialitása vagy osztály azonosítója, férőhelyek száma, és egy 5x8 mátrix].
- teacherMatrix: Itt a mátrix egy eleme [tanárok azonosítója, és egy 5x8].
- classMatrix: Itt a mátrix egy eleme [osztály azonosító, és egy 5x8 mátrix].
- teacher: A tanárok kívánságai.

Kimenet:

- subject: A bemeneti subject ellátva, hogy melyik időpontban van.
- roomMatrix: Ugyanaz, mint a bemenet csak jelölve vannak azok az időpontok, ahova tárgyat osztottunk.
- teacherMatrix: Ugyanaz, mint a bemenet csak jelölve vannak azok az időpontok, ahova tárgyat osztottunk.
- classMatrix: Ugyanaz, mint a bemenet csak jelölve vannak azok az időpontok, ahova tárgyat osztottunk.

Feladat: Először a kívánságok alapján megpróbáljuk beosztani az órákat. Az időpontokat leellenőrizzük a roomMatrix, teacherNotFixMatrix, classMatrix mátrixok mátrix részében és ha jó, akkor beosztjuk oda az órát. Ha a kívánságok alapján nincs megfelelő időpont, akkor feltöltjük a maradék helyeket.

lowerclassTimetableCreator

Bemenet:

- user: A felhasználók adatai és óráik.
- names: A T_NAMES tábla adatai.
- subject: Az alsós tárgyak.
- room: A termek adatai.
- RoomMatrix: Egy mátrix, aminek egy eleme [szoba azonosító, specialitása vagy osztály azonosítója, férőhelyek száma, és egy 5x8 mátrix].
- teacherMatrix: Itt a mátrix egy eleme [tanárok azonosítója, és egy 5x8].
- ClassMatrix: Itt a mátrix egy eleme [osztály azonosító, és egy 5x8 mátrix].
- teacher: A tanárok kívánságai.
- years_id: A generált év azonosítója.

Kimenet:

- lowerClassTimetable: A kigenerált órarend adatai vannak a tömbben felsorolva.

Feladat: Végigmegyünk a subject tömbön és minden egyes tárgyhoz a roomMatrix, teacherNotFixMatrix, classMatrix mátrixokból kiválogatjuk az adatokat, amik a T_TIMETALE táblába kellene. Majd ezeket visszaadjuk egy nagy tömbként.

Órarend generáló függvény

Bemenete:

- years_id: Félév azonosítója.

Kimenet: Kész alsós órarend.

Feladat: Alsós órarend generálása.

Algoritmus: Az algoritmus struktogramja a 73. ábrán látható. Melynek rendes stuktogrammja a mellékletben a stukik/lowerclass.nsd helyen található. Ha nincs programunk akkor stukik/lowerclass.png képet is megtekinthetjük.

lowerClass(years_id)	
subject = selectSubject(years_id)	
teacher = selectTeacher(years_id)	
names = selectNames()	
room = selectRoom()	
user = selectUser()	
timetable = createtmetable(subject, room, names)	
isGoodRooms(timetable, subject, room)	
user = serachTeacher(timetable, teacher, user, subject, names)	
user = UserAddDreamClass(subject, teacher, user)	
user = userAddBadDreamClass(subject, user)	
subject, user = settingSubjectmatetoFirstClass(subject, user)	
user	
teacherMatrix.add([user.id, [[0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0], [0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0]])	
LowerClassId	
classMatrix.add([classId, [[0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0], [0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0]])	
room	
<div> <div> <div></div> <div>room.fix ≠ 0</div> <div></div> </div> <div> <div>T</div> <div></div> <div>F</div> </div> </div> <div> <div>roomMatrix.add([room.Id, room.Fix, room.size, [[0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0], [0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0]])</div> <div>roomMatrix.add([room.Id, room.class, room.size, [[0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0], [0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0]])</div> </div>	
subject, rooMatrix, teacherMatrix, classMatrix = creatTmetable(user, subject, rooMatrix, teacherMatrix, classMatrix, teacher)	
lowerclassTimetable = lowerclassTimetableCreator(subject, classMatrix, teacherMatrix, RoomMatrix, names, room, user, years_id)	
lowerclassTimetable	
insertTimetableToTTimetable(lowerclassTimetableRow)	

73. ábra -lowerClass stuktogram

Felső órarend

Algoritmus rövid leírása

Az algoritmus az adatbázisból a felsős adatok lekérésével kezdődik. Először párosítjuk azokat az órákat, akiket csoportosítva szeretnénk tartani. Ezután ketté szedjük a speciális és nem speciális termet igénylő órákat. Majd minden osztálynak, akiknek nincs kijelölve osztályterem, az algoritmus kijelöl egy osztálytermet. Összepárosítjuk a tanárokat és az órákat. Végül az előzőhöz hasonlóan időpontokat keresünk az óráknak és kigeneráljuk az órarendet.

Megvalósítandó függvények

Előfordulhatnak olyan függvények, amiket az alsós órarend generálásnál már alkalmaztunk és itt is újra használjuk őket. Ezek nem lesznek újra implementálva.

Adatok szűrése

initializeUpperVariable

Bemenet:

- actualUpperClass: Az aktuális felsős osztályok azonosítói.
- room: A termek adatai.
- user: A tanárok listája.
- names: A T_NAMES adatai.

Kimenet:

- room: Az alsós termek nélküli termek listája.
- user: Alsós tanár nélküli tanárok listája.

Feladat: Visszaadja a felsős termeket és tanárokat.

initMateSubjectList

Bemenet:

- subject: Felső tárgyak.

Kimenet:

- subject: Az összevont óráknak a listája (Tehát akinek van subjectmate_id-ja).

Feladat: Kilistázza azokat az órákat, akiknek vannak társórái.

initSpecANdNotSpecSubject

Bemenet:

- subject: Felső tárgyak.
- mateSubjectlist: Az összefüggő órák párba állítva.

Kimenet:

- specSubject: A speciális terem igényű órák.
- normalSubject: A speciális terem igény nélküli órák.

Feladat: Kettébontja a tárgyakat speciális terem igényű órákra és a speciális terem igény nélküli órákra.

Adatok párosítása

selectinClassroomToClass

Bemenet:

- actualUpperClass: Az aktuális felsős osztályok azonosítói.
- room: A termek adatai.
- normalSubject: A speciális terem igény nélküli órák.

Kimenet:

- room: A terem adatok osztály párosítással.

Feladat: Amelyik osztálynak nincs terme, annak kijelölünk egy osztálytermet.

searchSubjectAndTeacherPairCombo

Bemenet:

- mateSubjectList: A társórák listája.
- teacher: A tanárok kívánságai.
- user: A tanárok listája.

Kimenet:

- mateSubjectList: A társórák listája tanár párosítással.

Feladat: A társórákhoz tanárt keresni először a kívánságok szerint. Ha nem találunk megfelelő időpontot a kívánságok között, akkor a maradék időpontok közül választunk megfelelőt.

searchSubjectAndTeacherPairSimpla

Bemenet:

- subject: A társóra nélküli órák listája.
- teacher: A tanárok kívánságai.
- user: A tanárok listája.

Kimenet:

- subject: Az órákhoz tanár párosítva.
- user: A felhasználók óra párosítással.

Feladat: Megpróbálunk a társórák nélküli óráknak tanárt keresni. Először a kívánságok szerint, aztán a maradék időpontokból válogatva.

Végső órarend összeállítása

insertMatrixToSubject

Bemenet:

- mateSubjectList: A társóra órák listája.
- teacher: A tanárok kívánságai.
- specSubject: A speciális tárgyak listája.
- normalSubjact: A specialitás nélküli órák listája.
- names: A T_NAMES adatai.
- roomMatrix: Egy mátrix, aminek egy eleme [szoba azonosít, speciális teremigény azonosító, osztály azonosítója akinek az osztályterme, férőhelyek száma, és egy 5x9 mátrix].
- userMatrix: Itt a mátrix egy eleme [tanárok azonosítója, és egy 5x9 mátrix].
- classMatrix: Itt a mátrix egy eleme [osztály azonosító, és egy 5x9 mátrix].

Kimenet:

- roomMatrix: Lefoglalva az órák időpontjait.
- userMátrix: Lefoglalva az órák időpontjait.
- classMatrix: Lefoglalva az órák időpontjai.

Feladat: Végig megyünk az órákon és először a kívánságokban kijelentett időpontokban próbáljuk meg lefoglalni az óra időpontját. Ha nem jó, akkor feltöltjük a maradék helyeket.

insertUpperClassToDb

Bemenet:

- subject: Az órák listája.
- names: A T_NAMES adatai.
- room: A termek listája.
- roomMatrix: Egy mátrix, aminek egy eleme [szoba azonosít, speciális teremigény azonosító, osztály azonosítója akinek az osztályterme, férőhelyek száma, és egy 5x9 mátrix].
- userMatrix: Itt a mátrix egy eleme [tanárok azonosítója, és egy 5x9 mátrix].
- classMatrix: Itt a mátrix egy eleme [osztály azonosító, és egy 5x9 mátrix].

Kimenet: A kigenerált órarend az adatbázisban van.

Feladat: Végig megy a mátrixokon és összeszedi belőle a generált adatokat, majd ezeket beszúrja az adatbázisba.

Órarend generáló függvény

Bemenete:

- years_id: Félév azonosítója.

Kimenet: Kész felsős órarend.

Feladat: Felsős órarend generálása.

Algoritmus: Az algoritmus struktogramja a 74. ábrán látható. Melynek rendes stuktogrammja a mellékletben a stukik/upperclass.nsd helyen található. Ha nincs programunk akkor stukik/upperclass.png képet is megtekinthetjük.

UpperClass(years_id)

```
subject = selectSubject(years_id)
```

```
teacher = selectTeacher(years_id)
```

```
names = selectNames()
```

```
room = selectRoom()
```

```
user = selectUser()
```

```
subject
```

```
    set(actualUpperClas).add(subject.class_id)
```

```
room, user = initializeUpperVariable(actualUpperClass, room, names, user)
```

```
mateSubjectList = initMateSubjectList(subject)
```

```
specSubject, normalSubject = initSpecANdNotSpecSubject(subject, mateSubjectList)
```

```
room = selectinClassroomToClass(actualUpperClass, room, normalSubject)
```

```
mateSubjectList = creatMateSubjectList(mateSubjectList, subject)
```

```
mateSubjectList, user = searchSubjectAndTeacherPairCombo(mateSubjectList, user, teacher)
```

```
specSubject, user = searchSubjectAndTeacherPairSimpla(specSubject, user, teacher)
```

```
normalSubject, user = searchSubjectAndTeacherPairSimpla(normalSubject, user, teacher)
```

```
user
```

```
    userMatrix.append(  
        [user.id, [[0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0],  
        [0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0]]])
```

```
actualUpperClass
```

```
    classMatrix.append(  
        [actualUpperClass.id, [[0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0],  
        [0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0]]])
```

```
room
```

```
    RoomMatrix.append([room.id, room.spec, room.class_id, room.size,  
        [[0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0],  
        [0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0]]])
```

```
classMatrix, userMatrix, RoomMatrix = insertMatrixToSubject(mateSubjectList, teacher, classMatrix, userMatrix,  
    RoomMatrix, specSubject, normalSubject, names)
```

```
    Ø
```

```
insertUpperClassToDb(subject, classMatrix, userMatrix, RoomMatrix, names, room)
```

74. ábra - Felső órarend stuktogramja

Frontend tervezete

Programozási nyelv választása

A felületre két nyelv volt, ami felmerült bennem. Az első az Angular, de ezt hamar elvettem, mert kis gondolkodás után rájöttem, hogy az egész oldal statikus és csak értelmetlenül bonyolítaná a programot. Ezért a PHP mellett döntöttem. Azon belül is a Yii2-es keretrendszerre esett a választásom mivel azt jól ismerem és az elképzelt felület megvalósításra tökéletesnek tűnt.

Felület terve

A program teljes felület terve megtalálható a mellékletben wireframe/egsodusWireframe.bmpr helyen. Ami balsamiq programmal létrehozott drótváz. A megtekintéséhez szükség van a balsamiq programra. Mivel elképzelhető, hogy nem rendelkezünk ilyen programmal ezért a dokumentációban lesznek felület képek és wireframe/kepek mappában png formátumban is megtalálható.

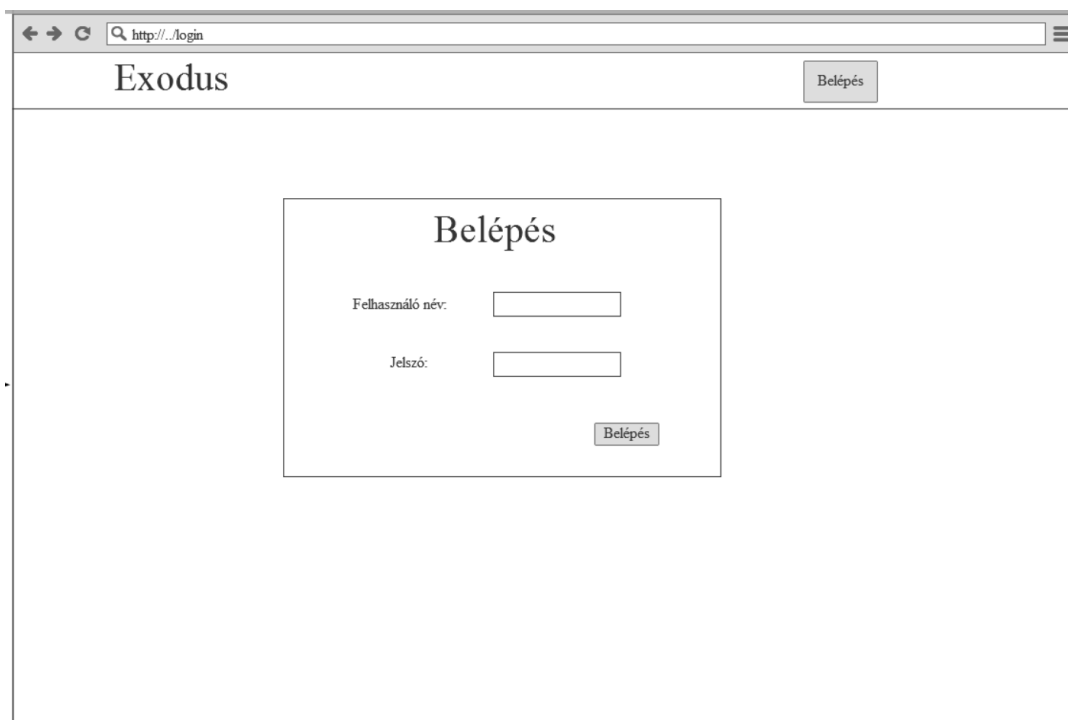
Felület tervezet működése

Ha nincsen balsamiqunk, hogy megnézzük a drótvázat, akkor itt egy kis leírás a drótvázról.

Ha megnyitjuk az alkalmazást, akkor a 75. ábrán látható oldal fog fogadni minket, ahonnan az ábrán látható belépés gombbal mehetünk tovább, ami a 76. ábrára továbbít minket.



75. ábra - Üdvözlő felület



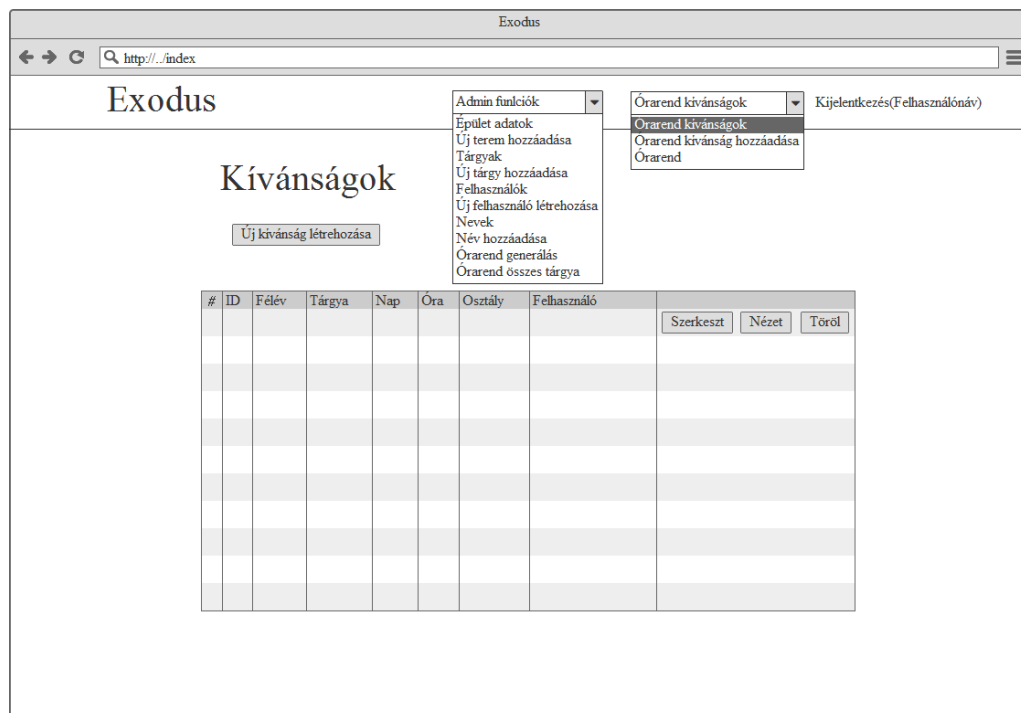
76. ábra - Belépés

Innen a jelszó alatti belépés gombbal jutunk tovább a 77. ábrára.



77. ábra - Index

A 77. ábrán látható gombok csak a drótváz használata miatt vannak a felületen. A normál felhasználóknak csak a funkciók menüsor lesz látható. Ha a 77. ábrán a kijelentkezés gombra kattintunk, akkor a 76. ábrára leszünk továbbítva. Ha az órarend kívánságokra kattintunk, akkor a 78. ábra fog bejönni.



78. ábra - Kívánságok

Ha innen az Új kívánság gombra vagy a fejlécben az Órarend kívánság funkcióra kattintunk, akkor a 79. ábrára jutunk. Ha a nézetre kattintunk, akkor a 80. ábrára kerülünk, ha a szerkesztés gombra kattintunk, akkor a 81. ábra fog bejönni. Ha a törlésre, akkor a 82. ábrára kerülünk. A 80. ábráról a Szerkesztéssel szintén a 81. ábrára jutunk. A 80. ábrán a törlés gombra kattintva a 82. ábrára fog feljönni.

Exodus

Kívánság létrehozása:

Félév: 2018/19-1

Tárgy: Angol

Nap: Hétfő

Óra: 1

Osztály: 1.a

Felhasználó: User1

Mentés

Admin funkciók

- Épület adatok
- Új terem hozzáadása
- Tárgyak
- Új tárgy hozzáadása
- Felhasználók
- Új felhasználó létrehozása
- Nevek
- Név hozzáadása
- Órarend generálás
- Órarend összes tárgya

Órarend kívánság hozzáadása

Órarend kívánságok

Órarend kívánság hozzáadása

Órarend

Kijelentkezés(Felhasználónév)

79. ábra - Új kívánság

Exodus

Szerkeszt Töröl

ID	Félév	Tárgy	Nap	Óra	Osztály	Felhasználó
1	2018/19-1	Alsós-reál	Hétfő	1	1.a	user1

Admin funkciók

- Épület adatok
- Új terem hozzáadása
- Tárgyak
- Új tárgy hozzáadása
- Felhasználók
- Új felhasználó létrehozása
- Nevek
- Név hozzáadása
- Órarend generálás
- Órarend összes tárgya

Funkciók

- Órarend kívánságok
- Órarend kívánság hozzáadása
- Órarend

Kijelentkezés(Felhasználónév)

80. ábra - Kívánság nézet

Exodus

Admin funkciók

- Épület adatok
- Új terem hozzáadása
- Tárgyak
- Új tárgy hozzáadása
- Felhasználók
- Új felhasználó létrehozása
- Nevek
- Név hozzáadása
- Órarend generálás
- Órarend összes tárgya

Funkciók

- Órarend kívánságok
- Órarend kívánság hozzáadása
- Órarend

Kijelentkezés(Felhasználónév)

Kívánság szerkesztése:

Félév
2018/19-1

Tárgy
Angol

Nap
Hétfő

Óra
1

Osztály
1.a

Felhasználó
User1

Mentés

81. ábra - Kívánság szerkesztés

Törlés

Biztosan törölni szeretnéd?

ok Mégse

82. ábra - Törlés

Ha a teljes drótváz képeit egyesével ide raknám, akkor még 30-40 oldalon keresztül csak képekkel lenne tele a dokumentációm és mivel az Épületadatok, Tárgyak, Felhasználók, Nevek, Órarend összes tárgya oldalak és a hozzá tartozó mellék oldalak felépítése egy az egyben megegyezik a Kívánságok oldal felépítésével, ezért ezt nem fogom beletenni. Akit érdekel és nem tudta megnyitni a fájlt, annak a mellékletben a wireframe/képek mappában megtalálja a teljes drótvázat png formátumban ahogy korábban említettem. A

következő két ábra még érdekes a felületről mivel a 83. ábrán láthatjuk, hogy milyen a tervezett órarend lekérő/megjelenítő felület. A 84. ábrán pedig látszik az óragenerátor felület. Ezek a felületek nagymértékben különböznek az eddigiektől ezért szerepelnek itt.

The screenshot shows the 'Exodus' application interface. At the top, there's a navigation bar with the title 'Exodus' and a search bar. Below the navigation bar, there are several dropdown menus for filtering: 'Admin funkciók', 'Órarend', and 'Kijelentkezés(Felhasználónév)'. The 'Admin funkciók' dropdown is open, showing options like 'Épület adatok', 'Új terem hozzáadása', 'Tárgyak', 'Új tárgy hozzáadása', 'Felhasználók', 'Új felhasználó létrehozása', 'Nevek', 'Név hozzáadása', 'Órarend generálás', and 'Órarend összes tárgya'. The 'Órarend' dropdown is also open, showing 'Órarend kívánságok', 'Órarend kívánság hozzáadása', and 'Órarend'. The main content area has three filters: 'Félév' (17/18-1), 'osztály' (1.a), and 'Tanár' (17/18-1). Below these filters is a 'lekérés' button. The main content area also contains a table with the following structure:

	Hétfő	Kedd	Szerda	Csütörtök	Péntek
0. óra	terem-tárgy-osztály-tanár			terem-tárgy-osztály-tanár	terem-tárgy-osztály-tanár
1. óra	terem-tárgy-osztály-tanár			terem-tárgy-osztály-tanár	terem-tárgy-osztály-tanár
2. óra	terem-tárgy-osztály-tanár			terem-tárgy-osztály-tanár	terem-tárgy-osztály-tanár
3. óra	terem-tárgy-osztály-tanár	terem-tárgy-osztály-tanár		terem-tárgy-osztály-tanár	terem-tárgy-osztály-tanár
4. óra	terem-tárgy-osztály-tanár	terem-tárgy-osztály-tanár		terem-tárgy-osztály-tanár	terem-tárgy-osztály-tanár
5. óra		terem-tárgy-osztály-tanár	terem-tárgy-osztály-tanár	terem-tárgy-osztály-tanár	terem-tárgy-osztály-tanár
6. óra			terem-tárgy-osztály-tanár	terem-tárgy-osztály-tanár	terem-tárgy-osztály-tanár
7. óra			terem-tárgy-osztály-tanár		
8. óra			terem-tárgy-osztály-tanár		

83. ábra - Órarend

The screenshot shows the 'Exodus' application interface for the 'Órarend generálása' (Timetable Generation) page. The navigation bar is the same as in the previous screenshot. The main content area has a title 'Órarend generálása' and a filter 'félév:' (2018/19-1). Below the filter is a 'Generálás' button. The main content area also contains a table with the following structure:

	Hétfő	Kedd	Szerda	Csütörtök	Péntek
0. óra	terem-tárgy-osztály-tanár			terem-tárgy-osztály-tanár	terem-tárgy-osztály-tanár
1. óra	terem-tárgy-osztály-tanár			terem-tárgy-osztály-tanár	terem-tárgy-osztály-tanár
2. óra	terem-tárgy-osztály-tanár			terem-tárgy-osztály-tanár	terem-tárgy-osztály-tanár
3. óra	terem-tárgy-osztály-tanár	terem-tárgy-osztály-tanár		terem-tárgy-osztály-tanár	terem-tárgy-osztály-tanár
4. óra	terem-tárgy-osztály-tanár	terem-tárgy-osztály-tanár		terem-tárgy-osztály-tanár	terem-tárgy-osztály-tanár
5. óra		terem-tárgy-osztály-tanár	terem-tárgy-osztály-tanár	terem-tárgy-osztály-tanár	terem-tárgy-osztály-tanár
6. óra			terem-tárgy-osztály-tanár	terem-tárgy-osztály-tanár	terem-tárgy-osztály-tanár
7. óra			terem-tárgy-osztály-tanár		
8. óra			terem-tárgy-osztály-tanár		

84. ábra - Generálás

A 83. ábrán a legördülő listából ki tudjuk választani, hogy melyik osztály vagy tanár órarendjét kívánjuk látni és a lekérés gombbal be tudjuk tölteni. A 84. ábrán a generálás gombra kattintva új órarendet lehet létrehozni.

Felület megvalósítási terve

A felület a tervek szerint MVC struktúrában lesz megvalósítva a yii2 keretrendszer segítségével. A felület egy basic yii2-es applikációra (1) épül.

Adatbázis táblák MVC szerkezete

Egy MVC modell egy táblához fog tartozni. Ezek a modellek nagyon hasonlóan épülnek fel, ezért először általában mutatom be a szerkezetüket. A különböző táblákat ábrázoló modelleknél csak a különbségeket fogom kihangsúlyozni.

Modell

A modell fogja tartalmazni az adatbázis kapcsolatot. A modell két fájlból fog állni, amelyek a models mappában lesznek. A legtöbb esetben az első modell nevét az általa reprezentált tábla nevéből fogjuk származtatni úgy, hogy a tábla előtagjából a t betűt elhagyjuk és nagy kezdőbetűvel kezdjük. Például a T_BUILDING táblának a modellje a Building lesz. Ez a modell a yii/db/ActiveRecordból (2) fog származni és a következő függvényeket fogja tartalmazni: tableName. Ez a modell által reprezentált táblának a nevét adja vissza. Lesz egy rules függvény is, ami a táblának az egyes mezőire vonatkozó megszorításokat írja le. Az attributeLabels fogja tartalmazni, hogy a modellben megjelenő értékeknek mi legyen a megjelenítendő neve. Ha a modell által reprezentált táblának vannak kapcsolatai másik táblákkal, akkor azoknak a tábláknak a modelljeivel ábrázolják a táblakapcsolatokat. A modell másik része a modelnévSearch-ből fog állni. Például a Building táblánál a BuildingSearch lesz. Ez a modell az első modellből fog származni és ez fogja definiálni az előző modellben történő kereséseket. Ennek a következő függvényei vannak: rules: Itt tudjuk megerősíteni és kibővíteni a felületen az egyes mezőkre való megszorításokat. A következő ascenerios, mely azért felel, hogy a getes hívások url-je sokkal egyszerűbb legyen. Az utolsó itt található függvény a search. Ez felel az adatok szűréséért, rendezésért, esetleg a táblák közötti kapcsolatok definiálásáért.

View

Ez a réteg felel a megjelenítéséért. A modellekhez tartozó view-k a views mappán belül a modell nevével megegyező mappában találhatóak. Minden modellhez fog tartozni egy index, ami a gridView-ból származó widget (3) osztály segítségével megjeleníti egy táblázatban az adott modell adatbázis táblájában található értékeket. Ezen kívül megjeleníti az egyes sorok megtekintését, szerkesztését, törlését segítő gombokat. Az index felületen látszani fog, hogy milyen adatokról van szó és egy gomb, ami egy új elem beszúrását segíti elő a táblázatba.

Minden modellnek lesz egy view nevű fájlja is, ami a modell egy elemének a megtekintésére szolgál. Ezt a DetailView-ból származó widget (4) osztály segítségével teszi meg. Ezen kívül a felületen lesz két gomb is, az egyik a megjelenített adatok szerkesztését, a másik a törlését teszi lehetővé.

Lesz egy _form nevű oldal, ahol szerkeszthető a táblázat egy sorának egy modellje vagy felvihető egy új modell, ami attól függ, hogy az update vagy a create nevű fájlon keresztül hívjuk meg. Ha az update fájlon keresztül hívjuk meg, akkor egy létező modell adatait tölti be és tudjuk szerkeszteni. Ha a create fájlon keresztül hívjuk meg a _form fájlt, akkor egy új modellt tudunk létrehozni.

Controller

A controller a felületen zajló folyamatokat írja le. Az úgynevezett üzleti logikát mutatja be. A controllers mappában találhatóak és modelController nevet viselik. Például a Buildingnél a BuildingController. Minden controller a yii-nek a base controller (5) osztályából származik. A következő alapfüggvényekkel rendelkeznek: behaviors, ami azért felel, hogy a különböző get-es post-os hívásokat szabályozza. Az actionIndex a saját modelljéhez tartozó viewban található index oldalt rajzolja ki a képernyőre. Az actionView a modell egy elsődleges kulccsal azonosított elemét rajzolja ki a hozzá tartozó viewban található view fájlban leírtaknak megfelelően. Az actionCreate egy új elem létrehozásának a folyamatát tartalmazza. Az actionUpdate egy elem szerkesztésnek a menetét írja le. Az actionDelete egy modell törlésének a folyamatát tartalmazza. A findModel a modell egy elemét azonosítja az elsődleges kulcs segítségével.

A felület táblamodelljei

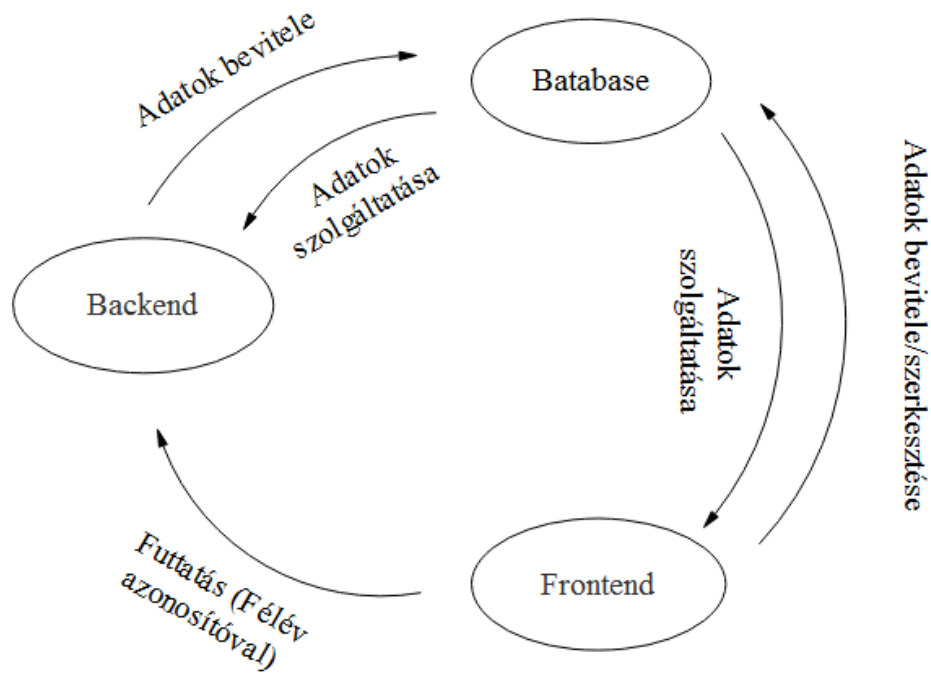
A T_BUILDING táblának a modellje a Building modell lesz. Itt az alap MVC-t nem kell kiegészíteni semmivel. A T_NAMES táblának a modellje a Names modell, amit szintén nem kell plusz függvénnyel bővíteni. A T_SUBJECT táblát a Subject modell fogja ábrázolni, ebben az esetben is szükségtelen a kiegészítés. A T_SUBJECT_TEACHER táblát a Subjectteacher modellben ábrázoljuk, aminek szintén nem lesz plusz függvénye. A T_USER táblát a Tuser modellben fogjuk ábrázolni. Ebben a modellben fogjuk implementálni az IdentityInterface-t (6). Ez az interfész szolgál a felhasználó kezelésért. A következő függvényeket kell hozzá felvinni: findIdentity, ami a felhasználót keresi meg a felhasználó neve alapján. FindByUsername, amely az esetünkben ugyanaz, mint a findIdentity. A getId visszaadja a felhasználó nevét. A validatePassword megvizsgálja, hogy a paraméterül kapott jelszó megegyezik-e a felhasználó jelszójával. A T_TIMETABLE táblát a Timetable modell írja le. Itt új folyamatokat kell implementálni a controllerben és új oldalakat kell tenni a viewban. A controllerben kell egy actionGenerate, amely betölti a felületen a view közül az generate view-t, aminek van egy legördülő listája, ahogy azt a 84. ábrán látjuk és egy generáló gomb melynek lenyomásával meghívjuk a backend python kódját a legördülő listában megadott félév azonosítójával. Ezen kívül szükségünk lesz egy másik függvényre a controllerben, amely az actionTimetable. Ez 83. ábrát tölti be, amely a view között timetable néven található meg. Alapesetben a bejelentkezett felhasználó órarendjét mutatjuk. Ha a lekérésnél kapunk egy osztályt és egy félév azonosítót, akkor a kapott osztálynak a paraméterben meghatározott félév óráit írjuk ki a képernyőre. Ha nem kapunk osztály azonosítót, de félév és felhasználó azonosítót igen, akkor az adott felhasználónak az adott félévi óráit mutatjuk meg. Minden más esetben a bejelentkezett felhasználó óráit jelenítjük meg.

Egyéb felületek

A bejelentkezés előtti felületeknek használjuk a basic yii applikációnak a felületét, amelyhez a modellek közül a loginForm (7) fog tartozni. A cotrollerek közül a siteController (8) szükséges. A view közül a site és a layouts fog kelleni.

Komponensek kommunikációja

A kommunikációs folyamatokat a 85. ábrán látható módon történnek. A frontend adatokat tud bekérni az adatbázisból és képes adatokat bevinni az adatbázisba. A felület fel tudja szólítani a backendet, hogy fusson le és generáljon órarendet. Erre a felhívásra a backend adatokat olvas az adatbázisból és a kész órarend adatai beviszi az adatbázisba.

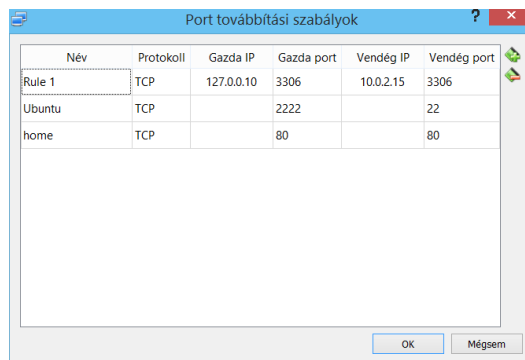


85. ábra Kommunikáció

Megvalósítás

Szerver

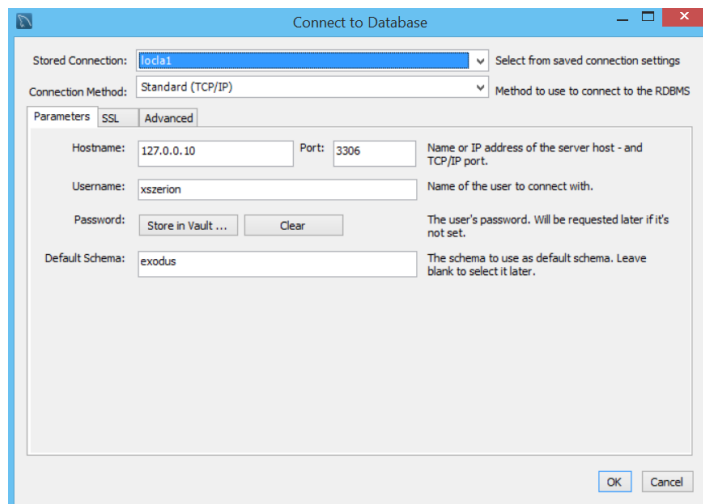
A szervernek egy ubuntu 18.04-et választottam, amit az Oracle VM virtuálBox-ra fel is telepítettem. Először brige-elt kártyát választottam hálózatnak. Ez sajnos a kollégiumban nem működött, így végül a Natolt hálózat mellett döntöttem. Beállítottam a 86. ábrán látható portokat, hogy elérje a felületet böngészőből és sftp-n keresztül. Plusz, hogy az adatbázist is elérjem. Mikor ez meg volt akkor Putty-on keresztül felcsatlakoztam a szerverre és a telepítés rendszerkövetelményeknél megtalálható programokat feltelepítettem és beállítottam az adatbázist. Szerencsére ez a folyamat hiba mentesen zajlott.



86. ábra - Portok

Adatbázis megvalósítás

Az adatbázist létrehozása a MySQL workbench segítségével történt. Mivel a felület tervét ebben a programban hoztam létre, ezért csak fel kell csatlakoznom a szerverre és kigenerálnom a programot. Ezt a következőképpen tehetjük meg: A ctrl+u billentyű kombinációval behozzuk a 87. ábrán látható felületet, majd kitöltjük a szerverünk adatait. De sajnos nem sikerült, ezért ctrl+shift+g-vel próbálkoztam, ami lehetővé teszi, hogy kiexportáljuk a felületről az SQL-t. A kiexportált SQL-t megpróbáltam a Toad MySQL-es verziójában lefuttatni. Sajnálatos módon ez is hibával elszállt. A gondot az okozta, hogy rossz verzió volt beállítva a szerveren. Átállítottam a MySQL workbenchben, hogy milyen verzióra generálja ki a táblákat és így már működött.



87. ábra - Adatbázis csatlakozás

Hamar rájöttem, hogy a Toad nagyon jó felületet nyújt az ORACLE adatbázishoz, de sajnálatosan nekem nem elég segítőkész a viszonylag ismeretlen MySQL-hez. Ezért guit váltottam és ettől fogva a HeidiSQL-t használtam az adatbázis szerkesztésére. A felületén a táblák ID mezőire egyszerűen alapértelmezett értéknek beállítottam az AUTO_INCREMENT értéket és innentől kezdve gyorsan létrejött. Majd elkezdtem felvinni az adatokat. Azt, hogy milyen tárgyak legyenek az edulinról (9) szedtem le. Ezután elkezdtem keresni és találtam egy alsós (10) és egy felsős (11) órarendet. Az adatfelvitel lassan haladt, ezért Excelben csináltam meg a tárgy adatokat és beimportáltam az adatbázisba. A felhasználók neveit a (12) oldalról vettem. Ezzel sikerült befejezmem az adatbázis feltöltését. Majd az adatokat változtatásával hoztam létre a teszteseteket.

Backend

A PHP stormot nagyon szeretem használni, ezért a PyCharm mellett döntöttem a sok hasonlóság miatt. Bár az egy kicsit aggályos, hogy 1,2 gigabyte memóriát elhasznál a gépből, de egyébként nagyon színvonalas és nagy segítséget jelent a fejlesztésben. Beállítottam az SFTP kapcsolatot a szerverre, így közvetlenül tudtam a szerveren fejleszteni. A programot Putty-on keresztül futtattam. Ennek az az előnye az Oracle VM virtual box-szal szemben, hogy itt működik a windozos felületre a másolás, így a hibákat könnyebb kideríteni. Az első feladat, amit meg kellett oldanom, az adatbázis csatlakozás volt, melyre a Python connector (13) csomagját használtam. A programozás további részében rájöttem, hogy a Python remekül jelzi a hibaüzeneteket. A fejlesztés következő problémája az alsós órarend összeállításánál jelentkezett. A tervben szerepeltek szerint,

együtt szerettem volna beosztani az órákat, de rájöttem, hogy hatékonyabb, ha a speciális termet nem igénylő órákat külön veszem a speciális igényűektől, így sokkal hatékonyabb és átláthatóbb kódot tudok generálni. Sikerült összeállítanom az alsós órarendet. Szinte hihetetlen, de 1 másodpercen belüli futás idővel rendelkezett. Végül ezeket az adatokat beszúrtam az adatbázisba, ekkor hirtelen meg tízszeresződött a program futási ideje. Ennek az volt az oka, hogy úgy szúrok be adatot az adatbázisba, hogy megnyitom az adatbázis kapcsolatot, beszúrok egy sort, majd bezárom és ez ismétlődik közel ötszázszor. Ezzel a módszerrel biztosabb, hogy minden adat bekerül az adatbázisba ezért e mellett döntöttem. A felsős órarend generálásánál nem igazán ütköztem problémába. A fejlesztés végén nekiálltam a hibakezelésnek, egyszerű printes hibaüzenetek helyett exception-eket használtam.

A fejlesztés során azt tetszett a legjobban, hogy szabadon szárnyalhattam és használhattam saját függvényeket is, amelyek közül a vég rekurzívan működő függvények tetszettek a legjobban.

Frontend

A frontend fejlesztéshez egyértelmű volt, hogy PHPStormot fogok használni, hiszen ez a legjobb gui a PHP-hoz. A PyChamból át is jöttek az SFTP adatok automatikusan ezért ennek a beállításával nem kellett törödnöm. A Putty-on keresztül felraktam a yii basic tamplet (14). Ezután beállítottam a config/db.php fájlba az adatbázis elérést és a config/web.php-ban átállítottam a gii-t, hogy a saját IP címemről tudjam elérni. De nem jött be a felület. Így felmentem a local szerverre és átraktam a teljes projektet a var/www/html mappába. Ez után már megjelent a felület. A gii (15) segítségével megpróbálta létrehozni a modelleket, de nem sikerült, mivel nem volt joga a szerveren létrehozni a fájlokat. A chmod 666-al megoldottam, hogy sikerüljön modellt generálni. Utána jöhetett a CRUD, de azzal nem sikerült létrehozni a controllert és a view-t sem. A chmod 666-tal a controllert sikerült megoldani, de a view-ba továbbra sem tudott beszúrni a felület. A view útvonala elé beraktam @ jelet és ez megoldotta a problémát. Ezután átállítottam az alkalmazás alapadatait és beállítottam a fejlécet. Majd igazítottam a felületen, hogy minden adat megfelelő helyen jelenjen meg. Megvalósítottam az órarend kiíratását. A php exec (16) függvényével megoldottam a php és Python közötti kommunikációt. Elvégeztem egy két biztonsági beállítás. Átállítottam az index oldalt és a logint, ezzel készen lettem a felülettel.

Összegzés

A program megvalósítása nagyon sok időbe és fáradtságos munkába került. Meglepő módon fele annyi hibába sem ütköztem, mint gondoltam és nagyon sok új hasznos tapasztalatot szereztem. Rájöttem, hogy a sikeres munkához alapos tervezés szükséges. Egy-egy hiba, ami körültekintő előkészítő munkával elkerülhető lett volna. Ezek a hibák a fejlesztés megvalósítása során többletmunkát eredményeztek. Arra is rájöttem, hogy csapatban mennyivel hatékonyabban tudok fejleszteni, mint egyedül. Korábban csapatban dolgoztam nagyobb projekteken, most pedig egyedül. A hibajavítás, egy-egy probléma megoldása, sokkal gyorsabb és egyszerűbb csapatban, mint egyedül. Végül nagyon sokat sikerült fejlődnöm így örülök, hogy ezt a projektet csináltam meg.

Tesztelés

Rendszer teszt

Tesztesetek

A 85. ábrán látható folyamatokat kell letesztelni.

A backend és adatbázis kommunikációja

Lekérés az adatbázisból

SelectSubject függvény

Teszteset: Olyan félét adunk meg paraméternek, ami nem létezik az adatbázisban.

Eredmény: Hibaüzenet: Nincsenek tárgyak az adott félévben!

Teszteset: Olyan félét adunk meg paraméternek, ami létezik az adatbázisban.

Eredmény: Visszaadja az paraméterben megkapott félév alsós tárgyait.

selectNames függvény

Teszteset: Meghívjuk.

Eredmény: Visszaadja a T_NAMES tábla értékeit.

selectTeacher

Teszteset: Olyan félét adunk meg paraméternek, ami nem létezik az adatbázisban.

Eredmény: Üres tömb.

Teszteset: Olyan félét adunk meg paraméternek, ami létezik az adatbázisban.

Eredmény: Visszaadja a paraméterben megadott félév alsós osztályaival kapcsolatok kívánságokat.

slectRoom

Teszteset: Meghívjuk.

Eredmény: Visszaadja a T_BUILDING tábla értékeit.

deletActualYears

Teszteset: Olyan félévet adunk meg paraméternek, ami nem létezik az adatbázisban.

Eredmény: Nem töröl semmit.

Teszteset: Olyan félévet adunk meg paraméternek, ami létezik az adatbázisban.

Eredmény: Kitörli a T_TIMETABLE táblából az aktuális félév adatait

selectTeacherUpper

Teszteset: Olyan félévet adunk meg paraméternek, ami nem létezik az adatbázisban.

Eredmény: Üres tömb.

Teszteset: Olyan félévet adunk meg paraméternek, ami létezik az adatbázisban.

Eredmény: Visszaadja a paraméterben megadott félév felsős osztályaival kapcsolatos kívánságokat.

selectSubjectUpper

Teszteset: Olyan félévet adunk meg paraméternek, ami nem létezik az adatbázisban.

Eredmény: Hibaüzenet: Nincsenek tárgyak az adott félévben!

Teszteset: Olyan félévet adunk meg paraméternek, ami létezik az adatbázisban.

Eredmény: Visszaadja az paraméterben megkapott félév felsős tárgyait.

Beszúrás az adatbázisba

insertTimetableToTTimetable

Teszteset: Nem jó struktúrát adunk meg paraméternek.

Eredmény: Hibaüzenet: String index out of range.

Teszteset: Jól strukturált adatsort adunk meg rossz adatokkal.

Eredmény: Hibaüzenet: Fail to delet record from datatbase: Incorrect integer value.

Teszteset: Jól strukturált adatsort adunk meg jó adatokkal.

Eredmény: Beszúrja az adatbázisba.

Frontend és backend függvényhívása

Teszteset: Rossz típusú paramétert adunk meg.

Eredmény: Hibaüzenet: Unknown colum.

Teszteset: Nem létező félév azonosítót adunk meg.

Eredmény: Hibaüzenet: Nincsenek tárgyak az adott félévben!

Teszteset: Létező félév azonosítót adunk meg adatok nélkül.

Eredmény: Hibaüzenet: Nincsenek tárgyak az adott félévben!

Teszteset: Létező félév azonosítót adunk meg adatokkal feltöltve.

Eredmény: Az órarend generálása kész.

Frontend és adatbázis kapcsolata

Adatbázisból való adatlekérés

Teszteset: Adatok lekérése.

Eredmény: Ha hibás adat van az adatbázisban, azt nem olvassa be. Így csak a modellnek megfelelő adatokat olvasunk be.

Adatbázisba való adat bevitel

Szerkesztés

Teszteset: Megpróbálunk az adatbázisba hibás adatokat felvinni.

Eredmény: A felület figyelmeztet a hibás értékekre.

Teszteset: Jó adatokra próbálunk meg beszúrni.

Eredmény: Módosítja az adatokat az adatbázisban.

Beszúrás

Teszteset: Megpróbálunk az adatbázisba hibás adatokat felvinni.

Eredmény: A felület figyelmeztet a hibás értékekre.

Teszteset: Jó adatokra próbálunk meg beszúrni.

Eredmény: Beszúrja az adatokat az adatbázisba.

Következtetés

A komponensek közötti kommunikáció nagyon jól működik. A hibakezelés is kielégítő, bár volt egy olyan eset, ahol nem egyértelmű a kapott hibaüzenet. Mikor nem létező félév azonosítóval hívjuk meg a backend kódot, akkor kaptunk ilyet. Lényegében megfogja a hibát és nem valószínű, hogy így hívjuk meg, ezért nem kell vele foglalkozni.

Modul teszt

A felület modul tesztjeit a felhasználói dokumentáció részben megteesszük. Az adatbázisban nincsenek függvények, ezért ott szükségtelen. Csak a backend részre kell modul teszteket végezni.

Backend tesztelése

Nem minden függvényt van értelme tesztelni. Ezen kívül vannak még függvények, amiket már máshol teszteltünk.

isGoodRooms

Teszteset: Timetable tömböt adunk meg bemenetnek, ahol az egyik felhasználó 53-szor van benne.

Eredmény: Hibaüzenet: A user1-nak/nek túl sok órája van.

Teszteset: Timetable tömböt adunk meg bemenetnek, ahol az egyik terem 53-szor van benne.

Eredmény: Hibaüzenet: A Tan terem túl van terhelve.

Teszteset: Egyik terem és felhasználó sem szerepel 39-nél többször.

Eredmény: Elfogadja a tömböt.

createFixOrNotFixUser

Tesztelés: A user adatoknál adunk meg mindenféle felhasználót.

Eredmény: Ketté szedi a tömböt a fix tanárookra és a speciális órákat tartókra.

notFixUserAddDreamClass

Teszteset: Üres a kívánságok tömbje.

Eredmény: Nem csinál semmit.

Teszteset: Vannak kívánságok benne, de vannak ugyanarra az órára is kívánságok több felhasználótól.

Eredmény: A tömbben az első kívánságot fogja figyelembe venni.

Teszteset: A kívánságok tömbjében nincs ütközés és minden tanár csak egy órát szeretne tartani.

Eredmény: Mindenkinek a kívánsága teljesül.

settingSubjectmatetoFirstClass

Teszteset: Kör van a tárgyaknál. Két tárgy egymásra mutat társóráként.

Eredmény: Végig megy a tömbön és index hibával elszáll.

Teszteset: Nincsenek társórák.

Eredmény: Nem csinál semmit.

Teszteset: Szabályos társórák vannak.

Eredmény: Csoportosítja a társórákat.

Következtetés

A program moduljai jól működne, a váratlan és hibás bemenetekre is az elvárt eredményt hozzák.

Terheléses teszt

A program egy iskolára készült. Ha nagyon nagy az iskola, akkor akár 150 felhasználóval is rendelkezik. A felületen zömmel, az adatbázissal végeznek műveleteket. A yii2 keretrendszer pedig az egyik legerősebb keretrendszer ilyen szempontból. Ez a dokumentációjában (17) is szerepel, így a felület akár több ezer felhasználót is elbír. A szerveren a kiosztható session számtól függ a felhasználók száma. Ugyanakkor, ha sok felhasználó van, akkor sok óra is van. Emiatt nagyon sok időbe telik, hogy az órarend tervező programban lefusson az adatbázis beszúrás, ezért 200 felhasználónál többet nem bír el a rendszer. Az algoritmus futtatásánál idő túllépésbe fog futni a program, ha a felhasználók számával egyenesen arányosan növeljük az órák számát. Viszont a 200 felhasználón megfelelő mennyiségű, mert ebbe bármelyik iskola tanári gárdája belefér. Így a program a funkcióját ellátja.

Hatékonyság tesztelése

A tesztelés a program hosszas használatával észrevett hatékonysági hibákat tartalmazza.

Felület

A felületet általános felhasználó szempontjából nézve: A felhasználó kívánságainak keresése, szűrése, szerkesztése kielégítő. A törlésnél lehetne olyan funkció, hogy több elemet kijelölünk és egyszerre kitöröljük. A kívánság felvitelénél is praktikus lenne, ha akár napokat ki tudnánk jelölni, vagy egyszerre az összes általunk jónak vélt időpontot egy táblázatban be tudnánk jelölni. Esetleg úgy tudnánk megnyitni egy kívánságot, hogy a megnyitással duplikáljuk és a másolatában csak egy-két adatot állítunk át. Az órarend felület nagyon jól működik.

Admin felhasználó szempontjából nézve: Elég hatékony a teljes felület, csak az órák felvitelénél lehetne valamilyen más megoldást találni. Például egy osztálynak az összes óráját duplikálni lehessen és megadni, hogy milyen osztályra cserélje ki az aktuális osztályt. Majd ezeket egyesével lehet módosítani és az aktuális osztályra igazítani.

Algoritmus

Az algoritmus az elvártak szerint működik. A hatékonyságával előfordulhatnak kisebb problémák. Ha egy felhasználónak nagyon sok kívánsága van, akkor az eltorzítja az órarendet. Ezen kívül, ha alsóban valaki egyedül szeretne egy osztályt, az ő órarendje nagyon jó lesz, de a társtanáré már kevésbé. Az algoritmus nagyon jól működik „tömött” órarendek esetében, illetve ha nagyon kevés tanár van az iskolában. Sok tanár esetén is tökéletes a hatékonyság, de előfordulhat, hogy a felesleges tanárok nem kapnak egy órát sem.

A program jövője

A programot több irányba is fejleszthetjük. Ha a felülettel és az algoritmussal tovább foglalkozunk, akkor számtalan dologra alkalmas lehet. Ezek megvalósulhatnak kisebb fejlesztésekkel például, hogy a felületről az órarendet ki lehessen nyomtatni vagy a diákok számára egy új felhasználó szintet megjeleníteni, hogy lássák az órarendjüket. Ezen kívül még bele lehetne rakni, hogy egy felhasználónak melyik időpontok nem jók vagy kit nem szeretne tanítani. De lehetnek nagyobb fejlesztések is. Például az egyik ilyen lehetőség, amit már korábban is említettem a Python nyelv választásánál az a következő lenne: A felületen betehetnénk egy értékelő panelt, hogy mennyire jó az adott órarend a

felhasználóknak. Ők visszajelzést küldenének és ez alapján egy mesterséges intelligencia modul segítségével az adott iskolára szabhatnánk az órarendet. A programot kisebb átalakítással alkalmassá lehet tenni egyetemi vagy gimnáziumi órarend tervezőnek. Az alkalmazásba bevezethetnénk egyéb funkciókat, amivel pl. a program alkalmassá válhatna E-napló vezetésére is. Lehetne belső fórumot kialakítani a tanároknak, hogy megosszák egymással az óraterveiket vagy más hasznos anyagokat. Olyan irányba is tovább mehetünk, hogy ne csak egy iskolának legyen az órarend tervezője, hanem többnek is. Be lehetne vezetni a felületen a többnyelvűséget. Ezeken felül számos tovább fejlesztési irány jöhet még szóba.

Irodalomjegyzék

1. [Online] 2018. 11 15. <https://www.yiiframework.com/doc/guide/2.0/en/start-installation>.
2. [Online] 2018. 11 15. <https://www.yiiframework.com/doc/api/2.0/yii-db-activerecord>.
3. [Online] 2018. 11 15. <https://www.yiiframework.com/doc/api/2.0/yii-grid-gridview>.
4. [Online] 2018. 11 15. <https://www.yiiframework.com/doc/api/2.0/yii-widgets-detailview>.
5. [Online] 2018. 11 15.
<https://translate.google.hu/translate?hl=hu&sl=en&tl=hu&u=https%3A%2F%2Fwww.yiiframework.com%2Fdoc%2Fapi%2F2.0%2Fyii-base-controller&anno=2>.
6. [Online] 2018. 11 15. <https://www.yiiframework.com/doc/api/2.0/yii-web-identityinterface>.
7. [Online] 2018. 11 15. <https://www.yiiframework.com/extension/yiisoft/yii2-app-advanced/doc/api/2.0/common-models-loginform>.
8. [Online] 2018. 11 15. <https://www.yiiframework.com/extension/yiisoft/yii2-app-basic/doc/api/2.0/app-controllers-sitecontroller>.
9. [Online] [Hivatkozva: 2018. 9 6.]
http://eduline.hu/kozoktatas/2013/4/16/Kerettantervek_tantargyak_listaja_6QN3E
9.
10. [Online] [Hivatkozva: 2018. 10 10.]
<http://www.ottevenyiskola.sulinet.hu/index.php/alsosorarend>.
11. [Online] [Hivatkozva: 2018. 10 15.]
<http://www.hermantudaskozpont.hu/iskola/orarend>.
12. [Online] [Hivatkozva: 2018. 9 21.]
<http://www.vicceskalandok.eoldal.hu/cikkek/vicces-nevek.html>.
13. [Online] [Hivatkozva: 2018. 8 15.] <https://dev.mysql.com/doc/connector-python/en/>.

- 14. [Online] [Hivatkozva: 2018. 11 10.]**
<https://www.yiiframework.com/doc/guide/2.0/en/start-installation>.
- 15. [Online] <https://www.yiiframework.com/doc/guide/2.0/en/start-gii>.**
- 16. [Online] [Hivatkozva: 2018. 11 22.]**
<http://php.net/manual/en/function.exec.php>.
- 17. [Online] [Hivatkozva: 2018. 12 01.]**
<https://www.yiiframework.com/doc/guide/2.0/en/intro-yii>.